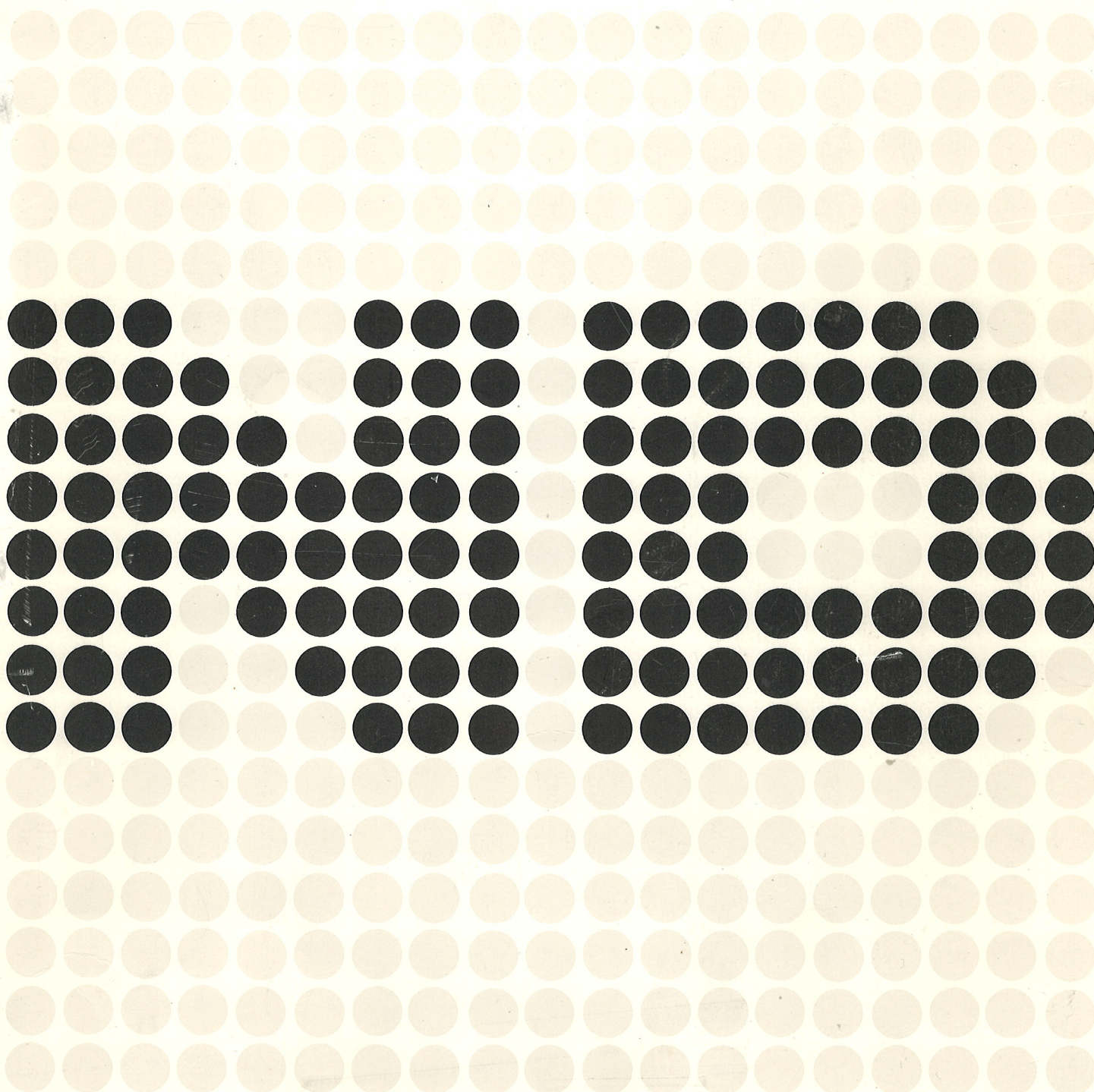


**SINTRAN III**  
**Reference Manual**

**NORSK DATA A.S**



# **SINTRAN III**

## **Reference Manual**

*NOTICE*

The information in this document is subject to change without notice. Norsk Data A.S assumes no responsibility for any errors that may appear in this document. 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.

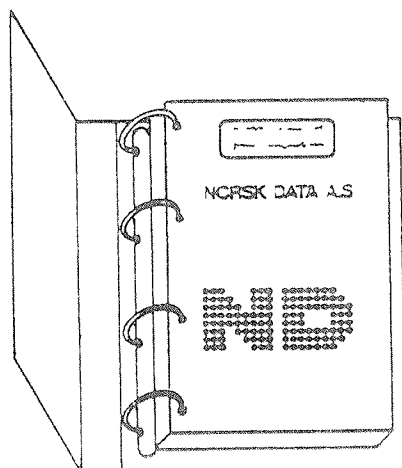
The information described in this document is protected by copyright. It may not be photocopied, reproduced or translated without the prior consent of Norsk Data A.S.

Copyright © 1981 by Norsk Data A.S.

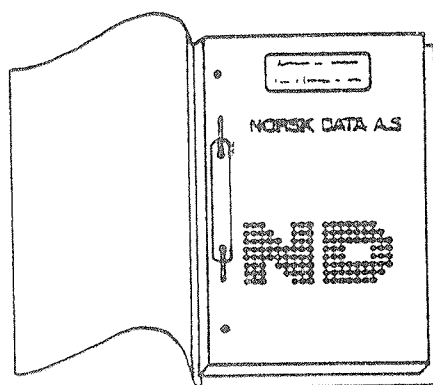
This manual is in loose leaf form for ease of updating. Old pages may be removed and new pages easily inserted if the manual is revised.

The loose leaf form also allows you to place the manual in a ring binder (A) for greater protection and convenience of use. Ring binders with 4 rings corresponding to the holes in the manual may be ordered in two widths, 30 mm and 40 mm. Use the order form below.

The manual may also be placed in a plastic cover (B). This cover is more suitable for manuals of less than 100 pages than for large manuals. Plastic covers may also be ordered below.



A Ring Binder



B Plastic Cover

Please send your order to the local ND office or (in Norway) to:

Documentation Department  
Norsk Data A.S.  
P.O. Box 4, Lindeberg gård  
Oslo 10

---

## ORDER FORM

I would like to order

..... Ring Binders, 30 mm, at nkr 20,- per binder

..... Ring Binders, 40 mm, at nkr 25,- per binder

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

Name .....

Company .....

Address .....

City .....



SINTRAN III Reference Manual  
Publication No. ND-60.128.02



NORSK DATA A.S  
P.O. Box 4, Lindeberg gård  
Oslo 10, Norway

Manuals can be updated in two ways, new versions and revisions. New versions consist of a complete new manual which replaces the old manual. New versions 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.

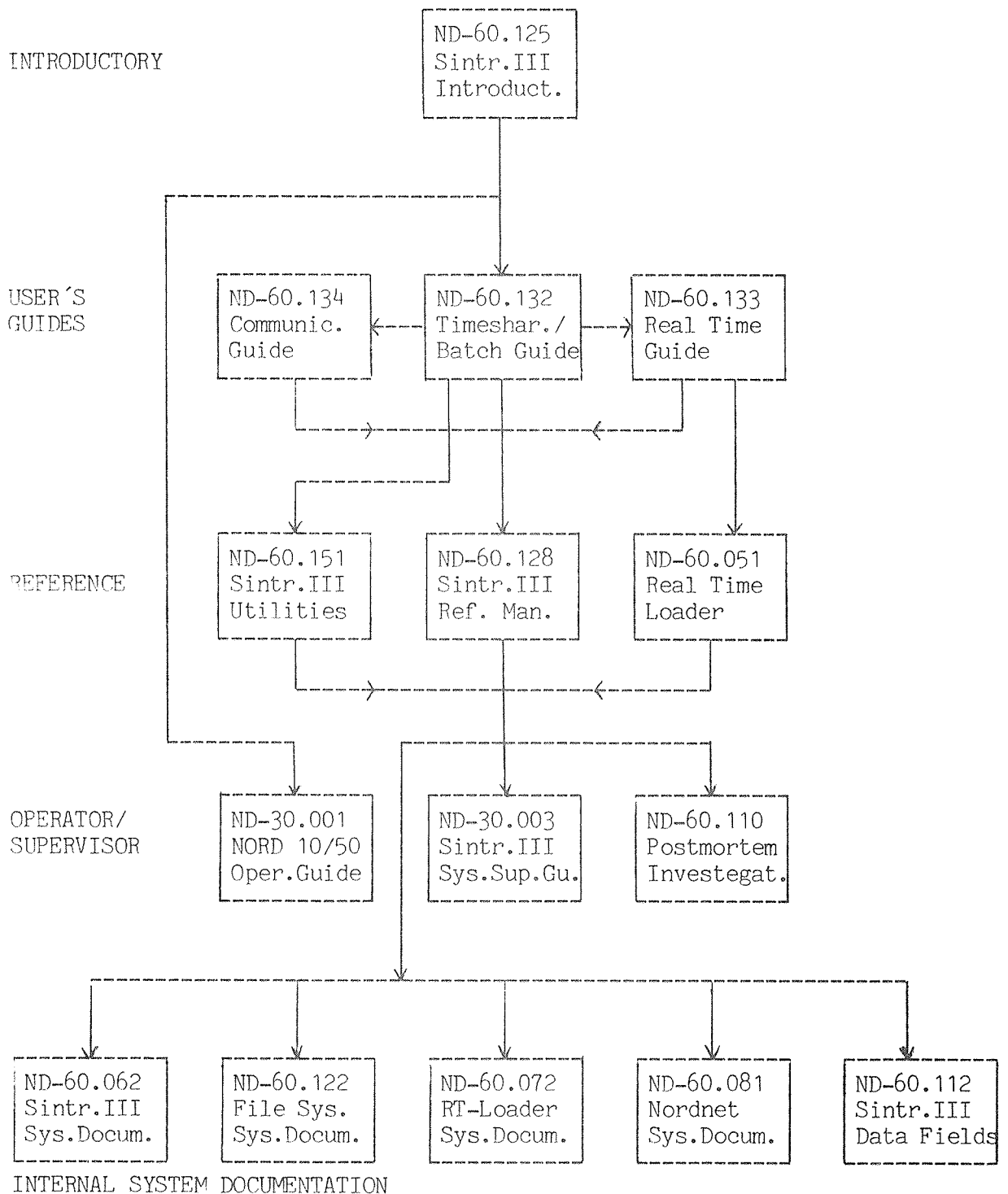
New versions and revisions are announced in the ND Bulletin and can be ordered as described below.

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

These forms, together with all types of inquiry and requests for documentation should be sent to the local ND office or (in Norway) to:

Documentation Department  
Norsk Data A.S  
P.O. Box 4, Lindeberg gård  
Oslo 10

## SINTRAN III/VS



## SINTRAN III/RT

ND-60.082  
Sin.III/RT  
Ref. Man.

## PREFACE

### *The Manual*

This manual describes the F version of SINTRAN III. The first part is an alphabetical list of the commands and their precise details.  
The second part is an alphabetical list of monitor calls and their details.  
The final part consists of appendices.

### *The Reader*

The manual is written for experienced users who need to look up precise details.

### *Related Manuals*

- SINTRAN III Introduction (ND-60.125)
- SINTRAN III Time-sharing/Batch Guide (ND-60.132)
- SINTRAN III Real-Time Guide (ND-60.133)
- SINTRAN III Communications Guide (ND-60.134)
- SINTRAN III System Supervisor (ND-30.003)



## HOW TO USE THE SINTRAN III DOCUMENTATION

The SINTRAN III documentation is written to satisfy the needs of a variety of users. It provides general introductory information for inexperienced users, precise detailed information for experienced users. Below are some questions a reader may have about how to make best use of the available documentation.

### *Question*

"I have no previous experience in using SINTRAN III. How do I find out what facilities are available in this operating system?"

### *Answer*

The SINTRAN III Introduction (ND-60.125) is a brief general introduction. It has simple examples which the user can perform on a computer terminal, providing first hand experience of SINTRAN III.

### *Question*

"I know the name of a command. Where do I find precise details of its function?"

### *Answer*

The SINTRAN III Reference Manual (ND-60.128), Part 1, lists the definition of all commands alphabetically by name.

### *Question*

"I know the name of a monitor call. Where do I find precise details of its function?"

*Answer*

The SINTRAN III Reference Manual (ND-60.128), Part 2, lists the definition of all monitor calls alphabetically by name.

*Question*

"I know the number of a monitor call. When do I find precise details of its function?"

*Answer*

The SINTRAN III Reference Manual (ND-60.128), Part 2, begins with a summary of monitor calls in numerical order. Find the name and look it up in the alphabetically ordered section.

*Question*

"I know the category of a command (or monitor call) but not its name. How do I find its name and information on how it is used?"

*Answer*

If it is a time-sharing command, find the category in the SINTRAN III Time-sharing/Batch Guide (ND-60.132) contents table. If it is a real-time command (reserved for user RT) use the Real-time Guide (ND-60.133) contents table. If it is a System Supervisor Command (reserved for user SYSTEM) consult the SINTRAN III System Supervisor Manual (ND-30.003).

*Question*

"I am looking for the name of a command (or monitor call) but I only know the name of a similar function. How do I find the name I am looking for?"

*Answer*

Look up the known name in the SINTRAN III Reference Manual (ND-60.128). Below the heading are names of similar or related functions.

## TABLE OF CONTENTS

+ + +

<i>Section:</i>	<i>Page:</i>
1	SINTRAN III COMMANDS (alphabetically) .....1—1
1.1	Introduction .....1—1
1.2	Command Summary .....1—3
1.3	Commands .....1—15
2	SINTRAN III MONITOR CALLS (alphabetically) .....2—1
2.1	Introduction .....2—1
2.2	Monitor Calls in Numerical Order .....2—2
2.3	Monitor Calls in Alphabetical Order .....2—5
<i>Appendix:</i>	
A	KEYBOARD LAYOUT OF STANDARD TERMINALS ..A—1
B	ND TERMINAL TYPES .....B—1
C	LOGICAL DEVICE NUMBERS USED IN SINTRAN III .....C—1
D	ERROR MESSAGES .....D—1
D.1	SINTRAN Monitor .....D—1
D.1.1	Run-Time Errors .....D—1
D.1.2	Run-Time Error Codes .....D—2
D.2	SINTRAN III File System .....D—5
D.2.1	Error Codes Returned from Monitor Calls .....D—5
E	STANDARD PERIPHERAL FILE NAMES .....E—1
F	STANDARD NAMES OF MASS STORAGE DEVICES .....F—1

*Appendix:**Page:*

G	ASCII CHARACTER SET .....	G—1
H	HARDWARE STATUS FOR MAGNETIC TAPE, CASSETTE TAPE, VERSATEC AND FLOPPY DISK ....	H—1
H.1	Status word for Tandberg, Pertec and STC magnetic tape units .....	H—1
H.2	Status word for Hewlett-Packard magnetic tape units .....	H—2
H.3	Status word for Phillips cassette .....	H—3
H.4	Status word for Versatec line printer/plotter .....	H—4
H.5	Status word for floppy disk .....	H—5
I	MONITOR CALLS AVAILABLE FROM FORTRAN ON ND-100 AND ND-500 .....	I—1



# 1 SINTRAN III COMMANDS (in alphabetical order)

## 1.1 INTRODUCTION

The command names are listed following the SINTRAN prompt character "@". This character is written to the terminal by the system when SINTRAN is ready to accept a command. The user types characters after the @.

Related commands and monitor calls are indicated. "See also" refers the reader to a command or monitor call with a related function. "See also related" refers the reader to a command or monitor call with a related function (eg. used in conjunction with this command).

A command can be written as one or several lines, in upper or lower case. If the user types carriage return (CR) before all parameters are entered, SINTRAN asks for the missing parameters on the next lines.

A command may be written in one of the following ways (user input is underlined):

@LIST-FILES ALFA,TERM

or

@LIST-FILES  
FILE NAME: ALFA  
OUTPUT FILE: TERM

or

@LIST-FILES,ALFA  
OUTPUT FILE: TERM

or

@LIST-FILES  
FILE NAME: ALFA,TERM

or

@LIST-FILES,ALFA,

The last example makes use of the default value for the parameter OUTPUT FILE. To use a default value for the last parameter in a command requires an extra comma; one comma causes a request for the parameter.

The following abbreviations used throughout:

CR = carriage return  
DEC = decimal number  
OCT = octal number  
DEF = default

The notation (OCT: 0 - 77) means an octal number in the range 0 to 77 inclusive.

The radix of a parameter may be specified by appending D (for decimal) or B (for octal) to the number.

Some parameters have default values; omitting parameters without default values causes an error message.

In some cases the notation [.....] is used to indicate that the parts in brackets may be omitted.

Parameters are separated by a comma or space. The SINTRAN editing characters are used to edit the current or previous command line.

Eg. pressing CTRL/A after

@LIST-FO

deletes the character O. If the command

@LIST-FI OLE,TERM

was the previous command entered, pressing CTRL/D causes this command to be reentered.

## 1.2 COMMAND SUMMARY

Command	Parameters	Used by	Short Description
ABORT	RT program name	RT	stop RT program
ABORT-BATCH	batch no.	RT	abort batch process
ABORT-JOB	batch no., user name	Public	abort current batch job
ABORT-PRINT	peripheral file	Public	abort current printout
ABSET	RT program name, seconds, minutes, hours	RT	start RT program at time of day
ALLOCATE-FILE	file name, page address, no. of pages	Public	create and allocate file
ALLOCATE-NEW-VERSION	file name, page address, no. of pages	Public	create and allocate new version of file
APPEND-BATCH	batch no., input file, output file	Public	append batch input/output files to batch queue
APPEND-REMOTE	host computer, input file	System	append remote batch input file to remote batch queue
APPEND-SPOOLING-FILE	peripheral file, file name, no. of copies, text [printing message]	Public	append file to spooling queue
BACKSPACE-PRINT	peripheral file, no. of pages, no. of lines	Public	backspace current print-out
BATCH	none	RT	start batch process
CC	none	Public	comment in batch or mode jobs
CHANGE-BACKGROUND-SEGMENT-SIZE	device no., no. of pages	System	change virtual memory size
CHANGE-BIT-FILE	directory name, block no.	System	change specified bit-file block
CHANGE-DIRECTORY-ENTRY	device name [,unit, fixed/removable, subunit]	System	change contents of directory entry
CHANGE-OBJECT-ENTRY	user name, object no.	System	change specified object entry

Command	Parameters	Used by	Short Description
CHANGE-PAGE	directory name, page no.	System	change specified page
CHANGE-PASSWORD	new password, old password	Public	change user password
CHANGE-USER-ENTRY	directory name, user no.	System	change user entry
CLADJ	no. of time units, time unit	RT	adjust internal clock
CLEAR-DEVICE	logical unit	System	execute a clear device (IOX) instruction
CLEAR-PASSWORD	user name	System	clear user password
CLOSE-FILE	file no.	Public	close opened file
COMMUNICATION-LINE-STATUS	line no.	Public	line status of a communication line
COMMUNICATION-STATUS	line no., output file	Public	line status of all communication lines
CONCT	RT program name, logical unit no.	RT	connect RT program to interrupt line
CONNECT-FILE	file name, file no., access mode	Public	open file for access through specified file number
CONTINUE	none	Public	restart background program
COPY	destination file, source file	Public	copy from/to file or device
COPY-DEVICE	destination device, source device	System	copy all pages from source device to destination device
COPY-DIRECTORY	destination directory name source directory name	System	copy all files from source directory to destination directory
COPY-FILE	destination file, source file	Public	copy file or device



Command	Parameters	Used by	Short Description
CREATE-DIRECTORY	directory name, device name [,unit, fixed/removable, subunit]	System	create a directory
CREATE-FILE	file name, no. of pages	Public	create one or more versions of a file
CREATE-FRIEND	friend name	Public	create another user as friend
CREATE-NEW-VERSION	file name, no. of pages	Public	create one or more new versions of a file
CREATE-USER	user name	System	create a new user
DATCL	none	Public	print current time and date on the terminal
DEFINE-ESCAPE-CHARACTER	logical unit, escape character	System	define value of break character
DEFINE-HISTOGRAM	program name, start address, interval	Public	define intervals for measurement of CPU time
DEFINE-SPOOLING-CONDITIONS	per file, printing name of spooling files?, stop and wait for START-PRINT, no. of lines per page	System	define conditions for spooling
DEFINE-SPOOLING-FILE-MESSAGE	user text	Public	define text to appear on the error device
DEFINE-SYSTEM-HISTOGRAM	program name, interrupt level, start address, interval	System	define address area and interrupt level for measurement of CPU time
DELETE-BATCH-QUEUE-ENTRY	batch no., input file, output file	Public	delete entry from batch queue
DELETE-FILE	file name	Public	delete file

Command	Parameters	Used by	Short Description
DELETE-FRIEND	friend name	Public	delete friend
DELETE-REENTRANT	subsystem name	System	delete reentrant subsystem
DELETE-REMOTE-QUEUE-ENTRY	host computer, queue entry	System	delete entry from remote batch queue
DELETE-SPOOLING-FILE	peripheral file, file name	Public	delete file from spooling queue
DELETE-USER	user name	System	delete user from directory
DELETE-USERS-FILES	file name, manual check	Public	delete all files with matching file names
DEVICE-FUNCTION	file name, function, optional parameter 1,2	Public	operates upon special devices
DIRECTORY-STATISTICS	directory name, output file	Public	list statistics of directories entered
DISABLE-ESCAPE-FUNCTION	logical unit	Public	set terminal in "non-user-break" mode
DMAC	none	RT	assembler for system debugging
DSCNT	RT program name	RT	disconnect RT program
DUMP	file name, start address, restart address	Public	save contents of users virtual memory
DUMP-BIT-FILE	directory name, block no., output file	System	octal dump of one 16 word block from bit-file of specified directory
DUMP-DIRECTORY-ENTRY	device name,[unit, fixed/removable, subunit,] output file	System	octal dump of directory entry
DUMP-OBJECT-ENTRY	user name, object no., output file	System	octal dump of object entry
DUMP-PAGE	directory name, page address, output file	System	octal dump of one 1K page
DUMP-REENTRANT	subsystem name, start address, restart address, binary file	System	establish subsystem as reentrant subsystem
DUMP-USER-ENTRY	directory name, user no., output file	System	octal dump of user entry

Command	Parameters	Used by	Short Description
ENABLE-ESCAPE-FUNCTION	logical unit	Public	reset "non-user-break" mode of terminal
ENTER	user name, password, project password, max. time	Public	identify the owner of a batch job
ENTER-DIRECTORY	directory name, device name, unit, fixed/removable, subunit	Public	enter directory into system
ENTSG	segment no., PT, interrupt level, start address	RT	initialize page table (PT)
EXECUTE-IOX	value, device no.	RT	execute IOX instruction
EXPAND-FILE	file name, no. of pages	Public	expand contiguous file by specified no. of pages
FILE-STATISTICS	file name, output file	Public	list statistics of files with matching names
FIX	segment no.	RT	fix segment in memory
FIXC	segment no., first physical page	RT	fix segment in contiguous memory area
FORWARD-SPACE-PRINT	peripheral file, no. of pages, no. of lines	Public	forward space current print-out
GET-ALTERNATIVE-RT-LOADER	file name	System	get backup of RT loader
GET-ALTERNATIVE-SYSTEM	file name 1, file name 2, is RT LOADER on segment file 0, name of RT LOADER file	System	load/start alternative SINTRAN III system
GET-ERROR-DEVICE	none	System	print logical unit number of device printing system errors
GET-RT-NAME	octal address	RT	convert address of RT description to RT program name
GET-TERMINAL-TYPE	logical unit	Public	write terminal type number
GIVE-SPOOLING-PAGES	no. of pages	System	increase no. of spooling pages
GIVE-USER-SPACE	user name, no. of pages	System	give user space in directory
GOTO-USER	octal address	Public	start background program
HELP	command, output file	Public	list matching commands
HOLD	no. of time units, time unit	Public	enter waiting state for specified time

Command	Parameters	Used by	Short Description
INIT-ACCOUNTING	desired, max., background[,RT, clear logged information, logging interval, ND-500]	System	initialize and start accounting of specified resources
INITIAL-COMMAND	command string	System	execute command at system restart
INITIALIZE- BACKGROUND- PROGRAMS	none	System	start background programs
INITIALIZE-ERROR-LOG	none	System	initialize error log segment
INTV	RT program name, no. of time units, time unit	RT	make RT program periodic
IOSET	Unit, input/output, RT program name, control	Public	reset device and clear device buffers
LIST-BATCH-PROCESS	none	Public	list status of batch processors
LIST-BATCH-QUEUE	batch no.	Public	list batch queue
LIST-DEVICE	logical unit, input/output	Public	list RT program reserving device
LIST-DEVICE- FUNCTIONS	output file	Public	list all functions available in @DEVICE-FUNCTION
LIST-DIRECTORIES- ENTERED	directory name, output file	Public	list names of directories entered
LIST-EXECUTION- QUEUE	none	Public	list execution queue
LIST-FILES	file name, output file	Public	list names of files with matching names
LIST-FRIENDS	friend name, output file	Public	list names of friends with matching names
LIST-INITIAL- COMMANDS	output file	System	list commands to be executed on system start
LIST-OPENED-FILES	output file	Public	list numbers/names of opened files
LIST-REENTRANT	none	Public	list reentrant subsystems
LIST-REMOTE-QUEUE	host computer	Public	list contents of remote batch queue



Command	Parameters	Used by	Short description
LIST-RT-DESCRIPTION	RT program name	Public	list RT description
LIST-RTOPENED-FILES	none	RT	list files opened by RT programs
LIST-RT-PROGRAMS	output file	RT	list RT program status
LIST-SEGMENT	segment no.	Public	list segment information
LIST-SPOOLING-FORM	peripheral file	Public	list key from SET-SPOOLING-FORM
LIST-SPOOLING-QUEUE	peripheral file, output file	Public	list spooling queue of a peripheral device
LIST-TIME-QUEUE	none	Public	list time queue
LIST-TITLE	none	Public	list system identification
LIST-USERS	user name, output file	Public	list all users with matching names
LOAD-BINARY	file name	Public	load and start programs in :BPUN format
LOCAL	none	Public	disconnect remote line
LOGOUT	none	Public	log out user and release terminal
LOOK-AT	space reference	Public	examine/change locations
MAIL	[output file]	Public	enter mail system
MAKE-ALTERNATIVE-RT-LOADER	file name	System	make backup of RT loader
MAKE-ALTERNATIVE-SYSTEM	destination SINTRAN file, MACM-AREA file, RT-loader file; source SINTRAN file, MACM-AREA file; is source RT loader in segment file 0, source RT loader file	System	make current SINTRAN III system into "alternative system"

Command	Parameters	Used by	Short Description
MEMORY	lower bound, upper bound	Public	define dump area
MODE	input file, output file	Public	accept SINTRAN commands from input file and write output to output file
MOVE-SPOOLING-QUEUE-ENTRY	peripheral file, file name, insert or append, before/after file name	Public	move specified spooling queue entry
NEXT-INITIAL-COMMAND	command string	System	specified command will be executed at restart
NORD-50	none	Public	start NORD-50 communication
OPCOM	none	System	enter operator's communication (ND-100, ND-100/500 only)
OPEN-FILE	file name, access mode	Public	open file for specified access
OPERATOR	text	Public	give message to operator
PLACE-BINARY	file name	Public	load :BPUN file into memory
PRINT-ERROR-LOG	output file	System	print error log segment
PRINT-HISTOGRAM	output file	Public	print percentage of CPU time spent in each interval defined in DEF-HIST
PRIOR	RT program name, priority	RT	set priority of RT program
PRLS	logical unit, input/output	RT	release unit from RT program
PRSRV	logical unit, input/output, RT program name	RT	reserve unit for RT program
RECOVER	file name	Public	start background program
REGENERATE-DIRECTORY	directory name	System	regenerate specified directory
RELEASE-DEVICE-UNIT	device name, unit	Public	release reserved device
RELEASE-DIRECTORY	directory name	Public	release directory
RELEASE-FILE	device name	Public	release logical unit
REMOTE	line number	Public	connect terminal to remote processor

Command	Parameters	Used by	Short Description
REMOTE-LOAD	load file, bootstrap address, line no.	RT	initialize remote load by bootstrap
REMOTE-PASSWORD	line no., password	System	set password for remote line from RT programs
REMOVE-FROM-SPOOLING-QUEUE	peripheral file, file name	Public	remove specified spooling queue entry
RENAME-DIRECTORY	old directory name, new directory name, device name [,unit, fixed/removable, subunit]	System	rename directory on specified device
RENAME-FILE	old file name, new object name:type	Public	change file name/type
RENAME-USER	old user name, new user name	System	change name of user in specified directory
RESERVE-DEVICE-UNIT	device name, unit	Public	reserve device for special use, directory cannot be entered
RESERVE-FILE	device name	Public	reserve specified unit for time-sharing user
RESTART-PRINT	peripheral file	Public	restart current printing
RESTART-SYSTEM		System	stop system and restart the system from mass storage
RESTART-USER	logical unit	RT	restart time-sharing user
RFILE	file no., memory address, block no., no. of words	Public	random read from file
RT	RT program name	RT	start RT program
RTCLOSE-FILE	file number	RT	close file for RT programs
RTCONNECT-FILE	file name, file no., access mode	RT	open file with number for access by RT programs
RTENTER	none	RT	allow file handling from RT programs
RT-LOADER	none	RT	load and start RT loader

Command	Parameters	Used by	Short Description
RTOFF	RT program name	RT	inhibit RT program
RTON	RT program name	RT	allow RT program to start
RTOPEN-FILE	file name, access mode	RT	open file for RT programs
RT-PROGRAM-LOG	RT program name, report interval, interrupts/sample, logical unit 1, logical unit 2, output file	RT	measure usage of computer resources
SAVE-DIRECTORY	none	System	same as @COPY-DEVICE
SCHEDULE	device no.	Public	reserve devices for batch job
SCRATCH-OPEN	file name, access mode	Public	open file as scratch file
SET	RT program name, no. of time units, time unit	RT	start RT program after a specified period
SET-AVAILABLE	none	System	set system available
SET-BLOCK-POINTER	file no., block no.	Public	set byte pointer to first byte in block
SET-BLOCK-SIZE	file no., block size	Public	set block size of opened file
SET-BYTE-POINTER	file no., byte no.	Public	set byte pointer to specified byte number
SET-DEFAULT-DIRECTORY	directory name	System	set specified directory as default directory
SET-DEFAULT-FILE-ACCESS	public access, friend access, owner access	Public	set default file access for all new files
SET-ERROR-DEVICE	logical unit	System	error messages appear on specified terminal
SET-FILE-ACCESS	file name, public access, friend access, owner access	Public	set access for specified file
SET-FRIEND-ACCESS	friend name, access mode	Public	set access mode of friend
SET-MEMORY-CONTENTS	contents, lower limit, upper limit	Public	set contents of user area
SET-NUMBER-OF-PRINT-COPIES	peripheral file, file name, no. of copies	Public	specify desired no. of copies
SET-PERIPHERAL-FILE	file name, device no.	System	define name of peripheral device

Command	Parameters	Used by	Short Description
SET-PERMANENT-OPENED	file no.	Public	open file permanently
SET-SPOOLING-FORM	peripheral file name, spooling form identification	System	define key for printing spooling file
SET-TEMPORARY-FILE	file name	Public	define file as temporary
SET-TERMINAL-FILE	file name	System	define name of terminal
SET-TERMINAL-TYPE	logical unit	Public	set terminal type number
SET-UNAVAILABLE	text	System	it is only possible to log in on terminal 1
SINTRAN-SERVICE-PROGRAM	none	System	activate the SINTRAN III service program
SPOOLING-PAGES-LEFT		Public	list number of spooling pages left
START-ACCOUNTING	background [,RT, clearing logged information, logging interval, ND-500]	System	start accounting specified resources
START-COMMUNICATION	line no.	RT	initiate communication on a line
START-HISTOGRAM	none	Public	start sampling of CPU time
START-PRINT	peripheral file	Public	resume printing of current printout
START-PROGRAM-LOG	interrupts/sample	System	start sampling of CPU usage
START-SPOOLING	peripheral file	System	start spooling for peripheral device
STATUS	none	Public	list register values
STOP-ACCOUNTING	background [,RT, ND-500]	System	stop accounting specified resources
STOP-COMMUNICATON	line no.	RT	terminate communication
STOP-HISTOGRAM	none	Public	stop sampling of CPU time
STOP-PRINT	peripheral file	Public	stop current printout
STOP-PROGRAM-LOG	output file	System	stop sampling and print result of CPU usage measurement

Command	Parameters	Used by	Short Description
STOP-SPOOLING	peripheral file	System	stop spooling for peripheral device
STOP-SYSTEM	none	System	simulate power failure
STOP-TERMINAL	logical unit	RT	log out user and release terminal
TAKE-SPOOLING-PAGES	no. of pages	System	decrease number of pages available for spooling files
TAKE-USER-SPACE	user name, no. of pages	System	take unused pages in directory
TERMINAL-MODE	capital letters, delay after carriage return, stop on full page, logout on missing carrier	Public	define operational requirements of a terminal
TERMINAL-STATUS	terminal no., interval	Public	print information about terminal
TEST-DIRECTORY	directory name	System	test directory for page conflicts
TIME-USED	none	Public	print time used
UNFIX	segment	RT	segment may be swapped
UPDAT	minute, hour, day, month, year	RT	update clock and calendar
USER-STATISTICS	user name, output file	Public	list statistics of named users
WAIT-FOR-OPERATOR	none	Public	wait for operator action
WFILE	file no., memory address, block no., no. of words	Public	random write to file
WHERE-IS-FILE	file name	Public	who has reserved device?
WHO-IS-ON	none	Public	who is logged in

## 1.3 COMMANDS

### @ABORT

See also ABORT (MON 105), @ABORT-JOB, @ABORT-PRINT and RTEXT (MON 134). See also related @STOP-TERMINAL.

**Function:**

Stops an RT program by setting it in the passive state. It is removed from the time queue and execution queue, all resources are released and periodic execution is discontinued.

**Format:**

@ABORT <program>

**Parameters:**

<program>

RT program name or RT description address  
(OCT, DEF = user's terminal background program).

**Rules:**

1. Permitted only for users RT and SYSTEM
2. The command has effect only if the program is in the running or wait state.
3. Do not abort a background program. If a background program is hanging, the system may have to be restarted. Contact your system supervisor.

**Example:**

@ABORT KLOKK

The RT program KLOKK is aborted.

### @ABORT-BATCH

See also related @ABORT-JOB and @BATCH.

**Function:**

Stop the batch processor by setting it in the passive state. Any currently running batch job is aborted and the batch queue cleared.

**Format:**

@ABORT-BATCH <batch no.>

**Parameters:**

<batch no.>

batch processor number as returned from @BATCH (DEC).

**Rules:**

1. Permitted only for user SYSTEM.
2. Effective only if the batch processor is idle or active.

**Example:**

@ABORT-BATCH 1

The batch processor no. 1 is aborted.

## @ABORT-JOB

See also @ABORT-BATCH.

### Function:

Abort the current batch job being processed. The next batch job in the batch queue (if any) will be initiated.

### Format:

@ABORT-JOB <batch no.>,<user name>

### Parameters:

<batch no.>

batch processor number as returned from @BATCH (DEC).

<user name>

owner of the batch job as specified by @ENTER.

### Rules:

1. A user may abort their own job and user SYSTEM may abort a batch job belonging to any <user name>.
2. Only effective if <user name> is logged in on the batch processor.

### Example:

@ABORT-JOB 2, GUEST

The current batch job for batch processor 2 is aborted if its owner is GUEST.

## @ABORT-PRINT

See also related @STOP-PRINT and @STOP-SPOOLING.

### Function:

Abort the current print-out on a spooling device and let the spooling program continue with the next file in the queue.

### Format:

@ABORT-PRINT <peripheral file name>

### Parameters:

<peripheral file name>

name of the spooling device.

### Rules:

1. Permitted only for user SYSTEM and the user who appended the file.
2. Effective only if the spooling program for the peripheral is started and a file is being printed on it.

### Example:

@ABORT-PRINT LINE-PRINTER

The current file on LINE-PRINTER is aborted.



**@ABSET**

See also ABSET (MON 102) and @SET.

**Function:**

Start an RT program at a specific time of day.

The program is put in the time queue and moved to the execution queue at the specified time.

**Format:**

@ABSET <program name>,<second>,<minute>,<hour>

**Parameters:**

<program name>

RT name or address of RT description (OCT, DEF = own terminal background program).

<second>

(DEC:0-59, DEF = 0).

<minute>

(DEC:0-59, DEF = 0).

<hour>

(DEC:0-23, DEF = 0).

**Rules:**

1. Permitted for users RT and SYSTEM.
2. If the time of day has passed, the program is scheduled for the next day.
3. If the program is already in the time queue it is removed and inserted according to the new specifications.
4. Programs in the time queue scheduled by @ABSET are rescheduled according to the new clock if the clock is adjusted (CLADJ).

**Example:**

@ABSET KLOKK, , 10, 13

KLOKK will be put in the execution queue at 13:10.

## @ALLOCATE-FILE

See also @CREATE-FILE and @ALLOCATE-NEW-VERSION.

### Function:

Create and allocate a contiguous file. The file is created on a specified area of a disk.

### Format:

@ALLOCATE-FILE <file name> , <page address> , <no. of pages>

### Parameters:

<file name>

optional version number specifies the number of versions allocated  
(DEF type = DATA, DEF version = 1).

<page address>

page number on the device where the file is to start (OCT: 1—).

<no. of pages>

size of the contiguous file area (DEC: 1—).

### Rules:

1. Permitted for all users.
2. If several versions are created, they are allocated one after the other.  
Version 1 will start at <page address>.
3. The allocated area must not be already in use.

### Example:

@ALLOCATE-FILE FILE-3, 100, 8

The contiguous file FILE-3:DATA is allocated in default directory starting at address 100<sub>8</sub>. Its size is 8<sub>10</sub> pages.

## @ALLOCATE-NEW-VERSION

See also @CREATE-NEW-VERSION and @ALLOCATE-FILE.

### Function:

Create and allocate a new version of a contiguous file. The file is created on a specified area of a disk.

### Format:

@ALLOCATE-NEW-VERSION <file name> , <page address> , <no. of pages>

### Parameters:

<file name>

for version (v) see rule 3 of @CREATE-NEW-VERSION (DEF type = DATA).

<page address>

<no. of pages>

see @ALLOCATE-FILE.

### Rules:

1. Permitted for all users with directory (D) access to <file name> .
2. The allocated area must not be already in use.

### Example:

@ALLOCATE-NEW-VERSION FILE-3;2,200,8

Version 2 of the contiguous file FILE-3 is created and allocated at address 200<sub>8</sub> in default directory. Its size is 8<sub>10</sub> pages.

## @APPEND-BATCH

See also @APPEND-REMOTE and @MODE. See also related @BATCH and @ENTER.

### Function:

Append batch job to the batch queue.

### Format:

@APPEND-BATCH <batch no.>, <input file>, <output file>

### Parameters:

<batch no.>

batch processor number as returned from @BATCH or @LIST-BATCH-PROCESS (DEC).

<input file>

name of a file containing one or more batch jobs (DEF type = SYMB).

<output file>

name of a file where the output is appended (DEF type = SYMB).

### Rules:

1. Permitted for all users.
2. <Input file> must have read access for all users with jobs on it and for the user SYSTEM.
3. <Output file> must have write append access for all users with jobs on the corresponding <input file>.
4. In a batch job, logical device number 1 means the <input file>/<output file> pair is in use.
5. The command is only valid if the batch processor is started (see @BATCH).

### Example:

@APPEND-BATCH 1, JOB-1, LINE-PRINTER

The batch-job-file JOB-1:SYMB is appended to batch processor 1. Output is appended to LINE-PRINTER.

## @APPEND-REMOTE

See also @APPEND-BATCH. See also related @LIST-REMOTE-QUEUE and @DELETE-REMOTE-QUEUE-ENTRY.

**Function:**

Append a batch input file to the remote batch queue for a remote computer. This is used for remote job entry (RJE) to a host computer.

**Format:**

@APPEND-REMOTE <remote computer>, <input file>

**Parameters:**

<remote computer>

peripheral file name denoting the host computer. Standard names are IBM, CDC, UNIVAC and HONEYWELL-BULL (DEF type = REM).

<input file>

name of the file containing one or more batch jobs.

**Rules:**

1. Permitted for all users.
2. <Input file> must have read access for user RT.

**Example:**

@APPEND-REMOTE UNIVAC, JOB-1

The batch-input-file JOB-1 is appended to the batch queue for the remote computer UNIVAC.

## @APPEND-SPOOLING-FILE

See also related @DEFINE-SPOOLING-CONDITIONS and @LIST-SPOOLING-QUEUE.

### Function:

Append one or more copies of a file to a spooling queue.

### Format:

@APPEND-SPOOLING-FILE <peripheral file name>,<file name>,<no. of copies>,<text>[,<printing message?>]

### Parameters:

<peripheral file name>

name of spooling device.

<file name>

the file to be appended.

<no. of copies>

number of copies to be output on the spooling device (DEC, DEF = 1).

<text>

any printable characters terminated by an apostrophe ('). The text is printed on the error device when file output is started (no text is specified by an ').

<printing message?>

YES = wait for a @START-PRINT command after printing <text>. This specification overrides @DEFINE-SPOOLING-CONDITIONS.

NO = the text is printed on the error device only if @DEFINE-SPOOLING-CONDITIONS specifies print (DEF = NO).

### Rules:

1. Permitted for all users.
2. If no <text> is specified, the last parameter is ignored.

### Example:

@APPEND-SPOOLING-FILE LINE-PRINTER,F-1,,MOUNT FORM-1',YES

One copy of the file F-1 is appended to the spooling queue to LINE-PRINTER. The message "MOUNT FORM-1" is output on the error device before the printing starts. The spooling program then waits for @START-PRINT.

## @BACKSPACE-PRINT

See also related @FORWARD-SPACE-PRINT, @STOP-PRINT, @START-PRINT and @RESTART-PRINT.

### Function:

Causes the spooling program to repeat the printing of the specified pages and lines in the current print file.

### Format:

@BACKSPACE-PRINT <peripheral file name>,<no. of pages>,<no. of lines>

### Parameters:

<peripheral file name>

<no. of pages>

<no. of lines>

amount of backspacing (DEC, DEF = 0).

### Rules:

1. Permitted only for user SYSTEM and the user who appended the current print file to the queue.
2. Valid only when spooling is started (@START-SPOOLING) and printing is in the stop-print state.
3. The number of lines per page may be changed by @DEFINE-SPOOLING-CONDITIONS.

### Example:

@STOP-PRINT LINE-PRINTER

@BACKSPACE-PRINT LINE-PRINTER,2,,

@START-PRINT LINE-PRINTER

The spooling output for LINE-PRINTER is stopped and output is resumed, starting at the beginning of the previous two pages.

**@BATCH**

See also related @ABORT-BATCH, @APPEND-BATCH and @LIST-BATCH-PROCESS.

**Function:**

Activate a passive batch processor.

**Format:**

@BATCH

**Parameters:**

None.

**Rules:**

1. Permitted only for users RT and SYSTEM.
2. The response is: BATCH NUMBER = n where n is a decimal number giving the batch processor number which identifies this batch processor in other commands.
3. If no passive batch processor is available an error message is given.
4. When a batch processor is started it enters the idle state because the batch queue is empty. It is activated by the first @APPEND-BATCH command.

**Example:**

@BATCH

BATCH NUMBER = 2

Batch processor no. 2 is activated. (Batch processor 1 is already active.)

**@CC****Function:**

Comment. The command has no effect. Normally used in batch jobs and MODE files.

**Format:**

@CC <text>

**Parameters:**

<text>

any printable characters.

**Rules:**

Permitted for all users.



## @CHANGE-BACKGROUND-SEGMENT-SIZE

See also ALTON (MON 33) and ALTOF (MON 34).

### Function:

Change segment size for a particular background program.

### Format:

@CHANGE-BACKGROUND-SEGMENT-SIZE <logical device no.>,<no. of pages>

### Parameters:

<logical device no.>

must be a terminal or batch processor (DEC).

<no. of pages>

must be 128 or 64 (DEC).

### Rules:

1. Permitted only for user SYSTEM.
2. This command must be executed after @INITIALIZE-BACKGROUND-PROGRAMS is issued.
3. A 64K segment is automatically reallocated if there is not enough space on the segment file for a 128K segment.

### Example:

@CHANGE-BACKGROUND-SEGMENT-SIZE 9,128

The background segment for terminal 9 is set to 128 pages.

**@CHANGE-BIT-FILE**

See also related @DUMP-BIT-FILE.

**Function:**

Change a block in the bit file of a directory.

**Format:**

@CHANGE-BIT-FILE <directory name>,<block no.>  
(subcommands)

**Parameters:**

<directory name>

<block no.>

see rule 3 below (DEC, DEF = 0).

**Rules:**

1. Permitted only for user SYSTEM.
2. The subcommands must follow the rules given in @LOOK-AT, rule 3. Terminate subcommands with a full stop (.).
3. The bit file is divided into 20<sub>8</sub> word blocks. <address> in subcommands is relative address within block (0 - 17<sub>8</sub>).

**Example:**

@CHANGE-BIT-FILE P-ONE,5

5/177777 157777

2000 .

@

## @CHANGE-DIRECTORY-ENTRY

See also related @DUMP-DIRECTORY-ENTRY.

### Function:

Change the contents of the directory entry on a device unit.

### Format:

@CHANGE-DIRECTORY-ENTRY <device name> [, <unit>, <'F' or 'R'>,  
<subunit> ]  
(subcommands)

### Parameters:

<device name>

See Appendix F.

<unit>

<'F' or 'R'>

<subunit>

see @CREATE-DIRECTORY.

### Rules:

1. Permitted only for user SYSTEM.
2. <unit> is only used if the device has more than one unit.
3. <'F' or 'R'> is only used for fixed or removable cartridge disk.
4. The subcommands must follow the rules given for @LOOK-AT. Terminate subcommands with a full stop (.). <address> is relative address in the range 0 - 17<sub>s</sub>.

### Example:

@CHANGE-DIRECTORY-ENTRY DISC-10MB-1,0,R

17/ 33 20

1000 .

@

Change the number of unreserved pages from 33<sub>s</sub> to 20<sub>s</sub>.

## @CHANGE-OBJECT-ENTRY

See also related @DUMP-OBJECT-ENTRY and @LIST-FILE.

### Function:

Change the specified object entry.

### Format:

@CHANGE-OBJECT-ENTRY [ <directory name>:] <user name>, <object no.>  
(subcommands)

### Parameters:

[ <directory name>:] <user name>  
<object no.>  
the object number of the file (OCT, DEF = 0).

### Rules:

1. Permitted only for user SYSTEM.
2. The subcommands must follow the rules given for @LOOK-AT, rule 3. Terminate subcommands with a full stop (.). <address> is relative address in the range 0 to 37<sub>8</sub>.

### Example:

@CHANGE-OBJECT-ENTRY P-T: GUEST, 15

35/ 1000 2000

2000 .

@

Change the maximum byte pointer in file no. 15 of user GUEST from 1000 to 2000.

**@CHANGE-PAGE**

See also related @DUMP-PAGE.

**Function:**

Change a page in a directory.

**Format:**

@CHANGE-PAGE <directory name> , <page address>  
(subcommands)

**Parameters:**

<directory name>

<page address>

address within the directory (OCT, DEF = 0).

**Rules:**

1. Permitted only for user SYSTEM.
2. The subcommands must follow the rules given for @LOOK-AT, rule 3. Terminate subcommands with a full stop (.). <address> is relative address within the page in the range 0 to 1777<sub>8</sub>.

**Example:**

@CHANGE-PAGE P-T, 100

400/ 1000 2000

2000

Ⓐ

Change location 400 on page 100 in directory pack three from 1000 to 2000.

## @CHANGE-PASSWORD

See also @CLEAR-PASSWORD.

### Function:

Change user password.

### Format:

@CHANGE-PASSWORD <old password> , <new password>

### Parameters:

<old password>  
(mandatory).  
<new password>  
(DEF = no password).

### Rules:

1. Permitted for all users.
2. The password is only changed if the old one is specified correctly.
3. A password may consist of printable and control characters, except carriage return.
4. The parameters are not echoed on the terminal if the command is entered on more than one line (see example).
5. Only the password of the logged in user may be changed.

### Example:

@CHANGE-PASSWORD

OLD PASSWORD: (old password is entered)

NEW PASSWORD: (new password is entered)

@

## @CHANGE-USER-ENTRY

See also related @DUMP-USER-ENTRY.

### Function:

Change a user entry in a directory.

### Format:

@CHANGE-USER-ENTRY <directory name>, <user no.>  
(subcommands)

### Parameters:

<directory name>

<user no.>

number of the user found by @USER-STATISTICS (DEC, DEF = 0).

### Rules:

1. Permitted only for user SYSTEM.
2. The subcommands must follow the rules given for @LOOK-AT, rule 3. Terminate subcommands with a full stop (.). <address> is relative address in the range 0 to 37<sub>8</sub>.

### Example:

@CHANGE-USER-ENTRY P-T, 19

20/ 1200 400

0

@

Change the number of pages used by user 19 in pack three from 1200 to 400.

## @CLADJ

See also CLADJ (MON 112) and @UPDAT.

**Function:**

Adjust internal clock.

**Format:**

@CLADJ <no. of time units>, <time unit>

**Parameters:**

<no. of time units>

time adjustment as a positive or negative number (DEC).

<time unit>

see @SET.

**Rules:**

1. Permitted only for user RT and SYSTEM.
2. The software clock and calendar are set forward if the adjustment is positive or stand still for the specified period if it is negative.
3. The time queue is checked and the schedule of all programs started by @ABSET is corrected.
4. On NORD-100 the command adjusts the panel clock if installed.

**Example:**

@CLADJ -25, 2

The internal clock will stand still for 25 seconds.



## @CLEAR-DEVICE

See also @EXECUTE-IOX and @IOSET.

### Function:

Execute a clear device operation (IOX instruction).

This command can eg. be used to stop the line printer if an attempt is made to print nonalphanumeric information, or it can be used to stop a search for a nonexistent EOF mark on a magnetic tape and prevent the tape from winding off.

### Format:

@CLEAR-DEVICE <logical device no.>

### Parameters:

<logical device no.>

logical number of a peripheral device (OCT).

### Rules:

1. Permitted only for user SYSTEM.
2. @CLEAR-DEVICE on a magnetic tape operates on the magnetic tape controller, ie. if several magnetic tape stations are connected to the same controller, they are all cleared through one @CLEAR-DEVICE command.

### Example:

@CLEAR-DEVICE 22

Clear the Versatec printer on DMA.

## @CLEAR-PASSWORD

See also @CHANGE-PASSWORD.

### Function:

Clear the password of a user.

### Format:

@CLEAR-PASSWORD <user name>

### Parameters:

<user name>

### Rules:

1. Permitted only for user SYSTEM.
2. User SYSTEM should not forget their password!

### Example:

@CLEAR-PASSWORD GUEST

The password for user GUEST is cleared.

## @CLOSE-FILE

See also CLOSE (MON 43) and SPCLO (MON 40). See also related @OPEN.

**Function:**

Close one or more files.

**Format:**

@CLOSE-FILE <file no.>

**Parameters:**

<file no.>

> 0: close the file

= -1: close all specified user files not permanently open

= -2: close all user files

(OCT)

**Rules:**

Permitted for all users.

**Example:**

@CLOSE 101

Close file number 101a.

## @ COMMUNICATION-LINE-STATUS

See also @COMMUNICATION-STATUS.

### Function:

Display short status list for a communication line. This command is similar to @COMMUNICATION-STATUS but <output file> is always **TERMINAL** and only items c to g (see @COMMUNICATION-STATUS) are listed.

### Format:

@COMMUNICATION-LINE-STATUS <line no.>

### Parameter:

<line no.>

(DEC, DEF = 1 if system is configured with only one line.)

### Rules:

1. Permitted for all users.
2. This command does not affect the communication.

### Example:

@COM-LI-ST

0 SEQUENCE ERRORS,	0 CHECKSUM ERRORS
0 NACKS SENT,	0 NACKS SENT SINCE LAST ACK SENT
0 NACKS RECEIVED,	0 FRAMES RETRANSMITTED SINCE ACK
	RECEIVED

34 FRAMES RETRANSMITTED SINCE COMMUNICATION STARTED  
COMMUNICATION RUNNING

@

## @COMMUNICATION-STATUS

See also @COMMUNICATION-LINE-STATUS

### Function:

List the following information for a communication line:

- a) "LOCAL DEVICE NUMBER", the line number on the other side of the channel.
- b) the logical device number for each channel on each side of the line. "BACKGROUND" means the channel has a background program.
- c) "NUMBER OF CONSECUTIVE FRAMES RETRANSMITTED ALLOWED".
- d) "NUMBER OF CONSECUTIVE NACKS SENT ALLOWED".
- e) "SEQUENCE ERRORS", messages received out of sequence.
- f) "NACKS SENT", frames retransmitted since last ack received.
- g) "NACKS RECEIVED", nacks received since last ack sent.
- h) "FRAMES RETRANSMITTED", the number of different frames retransmitted due to one or more NACK or timeout since communication started.
- i) COMMUNICATION RUNNING or COMMUNICATION DEAD.

### Format:

@COMMUNICATION-STATUS <output file> , <remote line no.>

### Parameters:

<line no.>

(DEC, DEF = 1 if system is configured with only one line.)

<output file>

destination of the list (DEF = TERMINAL).

### Rules:

1. Permitted for all users.
2. This command does not affect the communication.

### Example

@COM-STAT

OUTPUT FILE:

REMOTE LINE NUMBER: 1

```

CHANNEL  0:      LOCAL      DEVICE NUMBER = 600
                REMOTE      DEVICE NUMBER = 600 BACKGROUND
CHANNEL  1:      LOCAL      DEVICE NUMBER = 601 BACKGROUND
                REMOTE      DEVICE NUMBER = 601
CHANNEL  2:      LOCAL      DEVICE NUMBER = 602
                REMOTE      DEVICE NUMBER = 602 BACKGROUND
CHANNEL  3:      LOCAL      DEVICE NUMBER = 603 BACKGROUND
                REMOTE      DEVICE NUMBER = 603
CHANNEL  4:      LOCAL      DEVICE NUMBER = 604
                REMOTE      DEVICE NUMBER = 604 BACKGROUND
CHANNEL  5:      LOCAL      DEVICE NUMBER = 605 BACKGROUND
                REMOTE      DEVICE NUMBER = 605

NUMBER OF CONSECUTIVE FRAMES RETRANSMITTED ALLOWED:      4
NUMBER OF CONSECUTIVE NACKS SENT ALLOWED:                12
      0 SEQUENCE ERRORS,      0 CHECKSUM ERRORS
      0 NACKS SENT,           0 NACKS SENT SINCE LAST ACK SENT
      0 NACKS RECEIVED,      0 FRAMES RETRANSMITTED SINCE ACK
                                RECEIVED

34 FRAMES RETRANSMITTED SINCE COMMUNICATION STARTED
COMMUNICATION RUNNING
®
```

**@CONCT**

See also CONCT (MON 106). See also related @DSCNT.

**Function:**

Connect an RT program to the interrupt from a device. The RT program is put in the execution queue when the device gives an interrupt.

**Format:**

@CONCT <program name>, <logical device no.>

**Parameters:**

<program name>

RT program name or RT description address (OCT, DEF = user's terminal background program).

<logical device no.>

(DEC).

**Rules:**

1. Permitted only for user RT and SYSTEM.
2. Several units may be connected to one program.

**Example:**

@CONCT KLOKK, 410B

The RT description of KLOKK will be put in the execution queue every time an interrupt is given from device 410<sub>8</sub>.

## @CONNECT-FILE

See also @OPEN-FILE.

### Function:

Open a mass-storage file with a specified number.

### Format:

@CONNECT-FILE <file name> , <file no.> , <access type>

### Parameters:

<file name>  
     (DEF type = SYMB).  
 <file no.>  
     a logical device number (OCT: 100-121).  
 <access type>  
     see @OPEN-FILE.

### Rules:

1. Permitted for all users.
2. Valid only if file number has not been opened previously.
3. See @OPEN-FILE, rules 2, 4, 5 and 6.

### Example:

@CONNECT-FILE F-1, 120, R

File F-1:SYMB is opened for read and given file number 120<sub>8</sub>.

## @CONTINUE

See also related @GOTO-USER.

### Function:

Restart the execution of a program previously started by @RECOVER. The program is restarted at the <restart address> specified in @DUMP (or the \*DUMP subcommand in the NRL loader).

### Format:

@CONTINUE

### Parameters:

None.

### Rules:

1. Permitted for all users.
2. Cannot be used to restart:
  - (a) DMAC
  - (b) SINTRAN-SERVICE-PROGRAM
  - (c) MAIL
  - (d) RT loader
  - (e) a program started with PLACE and GOTO or LOAD.

**@COPY**

See also @COPY-FILE.

**Function:**

Copy data to a destination file from a source file. The file is copied byte by byte.

**Format:**

@COPY <destination file>, <source file>

**Parameters:**

<destination file>

a file to which data is copied. It may be any type of file (DEF type = SYMB).

<source file>

a file from which data is copied (DEF type = SYMB).

**Rules:**

1. Permitted for all users.
2. If the <source file> is a mass-storage file with a hole, the copying will stop at the hole and the error message NO SUCH PAGE is given.
3. If both <destination file> and <source file> are mass storage files, @COPY-FILE should be used.
4. Only the bytes up to the maximum byte pointer are copied from the <source file>.

## @COPY-DEVICE

See also @COPY-DIRECTORY.

### Function:

Copy all pages from the source (mass-storage) device to the destination device.

### Format:

@COPY-DEVICE <destination device name>,[<unit>],[<'F' or 'R'>],  
<source device name>[,<unit>][,<'F' or 'R'>]

### Parameters:

<destination device name>

<unit>

<'F' or 'R'>

see @CREATE-DIRECTORY.

<source device name>

<unit>

<'F' or 'R'>

see @CREATE-DIRECTORY.

### Rules:

1. Permitted only for user SYSTEM.
2. Valid only for devices which can contain directories.
3. A directory must exist on the source device.
4. Destination device must not be an entered directory.

### Example:

@COPY-DEVICE DISC-75MB-2,DISC-75MB-1

Copy all pages from one device to the other.



## @COPY-DIRECTORY

See also @COPY-DEVICE.

### Function:

Copy all files onto the destination directory from the source directory.

### Format:

@COPY-DIRECTORY <destination directory name>,<source directory name>

### Parameters:

<destination directory name>

<source directory name>

see @CREATE-DIRECTORY.

### Rules:

1. Permitted only for user SYSTEM.
2. The destination directory should have no users and no files when the command is given (use @CREATE-DIRECTORY).
3. The users and the file names are identical in both directories after the command is executed.
4. All files must be closed during the copying. Spooling should be turned off.

### Example:

@CREATE-DIRECTORY PACK-TWO,DI-33MB-2,,

@COPY-DEVICE PACK-TWO,PACK-ONE

Copy all users and all files in PACK-ONE onto PACK-TWO.

## @COPY-FILE

See also @COPY.

**Function:**

Copy data to a destination file from a source file. The file is copied page by page if both files are mass-storage files.

**Format:**

@COPY-FILE <destination file> , <source file>

**Parameters:**

<destination file>

a file to which data is copied (DEF type = SYMB).

<source file>

a file from which data is copied (DEF type = SYMB).

**Rules:**

1. Permitted for all users.
2. If the <destination file> is a peripheral file and the mass-storage <source file> is an indexed file with a hole, copying stops at the hole and the error message NO SUCH PAGE is given.
3. If both files are mass-storage files, the file is copied including holes.
4. All pages allocated to the <source file> are copied, except if there are zero bytes in the file.
5. If the <destination file> does not exist, it is created by giving the name in quotes; it will be an indexed file.

## @CREATE-DIRECTORY

See also related @ENTER-DIRECTORY.

### Function:

Create a new directory on a disk. The old directory is destroyed and all files are lost. When the command is executed, the directory name is written onto the first page of the device. The device is tested for bad spots (tracks) and a bit file is created with one bit for each page on the unit. This bit is set to one if the page is occupied. The bit file is normally allocated in the middle of the disk. This restricts the maximum length of a contiguous file to half the size of the disk. (The user can specify another location for the bit file.)

### Format:

@CREATE-DIRECTORY <directory name>,<device name>,[<unit>],[<'F' or 'R'>],[<subunit>],[<bit file address>

### Parameters:

<directory name>

the name to be written onto the new directory. A maximum of 16 alphanumeric characters, including the comma, are legal.

<device name>

the name of the mass storage device controller to which the device unit is connected. See Appendix F for legal names.

<unit>

the unit number. Only to be specified if SINTRAN III is generated with multiple units. Unit number starts at 0 (OCT, DEF = 0).

<'F' or 'R'>

F = a fixed cartridge disk pack. R = a removable cartridge disk pack.

<subunit>

applies to subdivided disk drives, eg. the big cartridge drive.

<bit file address>

specified if the user wants to place the bit file in a particular area. (OCT, DEF = the file system will select a medium dependent optimal value.) The System Supervisor manual contains more information on this parameter.

### Rules:

1. Permitted only for user SYSTEM when disk is specified, for all users when floppy disk.
2. Fixed or removable is specified only for 10, 30, 60, 90MBy cartridge disks.
3. If SINTRAN is configured with only one unit (number 0), the <unit> parameter (and comma) is omitted.

### Examples:

@CREATE-DIRECTORY PACK-ONE,DI-1-1,1,F

A directory named PACK-ONE is created on cartridge disk controller 1, unit 1, fixed pack.

@CREATE-DIRECTORY DIR-1,F-D-1,0

A directory with the name DIR-1 is created on the floppy disk controller 1, unit 0.

## @CREATE-FILE

See also @OPEN-FILE, @ALLOCATE-FILE and @CREATE-NEW-VERSION.

### Function:

Create one or more versions of a file. The file will be contiguous or indexed depending on <no. of pages>.

### Format:

@CREATE-FILE <file name>, <no. of pages>

### Parameters:

<file name>

optional version number specifies number of versions to be created (DEF type = DATA).

<no. of pages>

0: create empty indexed file

> 0: create contiguous file with the specified no. of pages  
(DEC, DEF = 0).

### Rules:

1. Permitted for all users.
2. If there are not enough pages for all versions, the system creates as many as possible and gives an error message. To find the number created, use @LIST-FILE.

### Example:

@CREATE-FILE F-1,0

One version of the indexed file F-1:DATA is created in default directory.

## @ CREATE-FRIEND

See also related @DELETE-FRIEND and @SET-FRIEND-ACCESS.

### Function:

Declare a user as a friend to the current user. This friendship is not reciprocal.

### Format:

@CREATE-FRIEND <user name>

### Parameter:

<user name>  
an existing user name.

### Rules:

1. Permitted for all users.
2. A user can have a maximum of eight friends belonging to the same main directory.
3. When the friend is created, his general file access to the terminal user's files is read, write and append (RWA). This can be changed by @SET-FRIEND-ACCESS. Friends can be given restricted access to a specific file by @SET-FILE-ACCESS.
4. Access given to a friend overrides public access, even if it is more limited than public access.

### Example:

@CREATE-FRIEND GUEST

The user GUEST is created as a friend to the terminal user and is given file access RWA.

## @CREATE-NEW-VERSION

See also @CREATE-FILE and @ALLOCATE-NEW-VERSION.

### Function:

Create one or more new versions of an existing file.

### Format:

@CREATE-NEW-VERSION <file name> , <no. of pages>

### Parameters:

<file name>

for version  $v$  see rule 3 below (DEF type = DATA).

<no. of pages>

see @CREATE-FILE.

### Rules:

1. Permitted for all users with directory access (D) to the file.
2. See @CREATE-FILE, rule 2.
3. If version  $v$  already exists, one new version is created and inserted as this version. The old version is renumbered as  $v + 1$  and so on. If  $v$  is higher than the highest version existing ( $= w$ ) the versions  $w + 1$ ,  $w + 2$ , ...,  $v$  are created. Default version is  $w + 1$ .
4. A new version can be created by other commands (eg. @OPEN-FILE). The file name is then enclosed in quotes.

### Example:

@CREATE-NEW-VERSION F-1;4,0

Versions 3 and 4 of the indexed file F-1 are created.

## @ CREATE-USER

See also related @DELETE-USER, @GIVE-USER-SPACE and @TAKE-USER-SPACE.

### Function:

Introduce a new user in a directory.

### Format:

@CREATE-USER [ <directory name>:]<user name>

### Parameters:

[...:]<user name>  
(DEF directory is main directory).

### Rules:

1. Permitted only for user SYSTEM when disk is specified, for all users when floppy disk.
2. A user must exist in a main directory before he is created in any other directory.
3. If the user needs space in the directory, @GIVE-USER-SPACE must be used.

### Example:

@CREATE-USER P-T:GUEST

The user GUEST is created in directory P-T.

**@DATCL**

See also TIME (MON 11) and CLOCK (MON 113). See also related @UPDAT.

**Function:**

Print the current setting of the clock and the date.

**Format:**

@DATCL

**Parameters:**

None.

**Rules:**

1. Permitted for all users.
2. The internal time is reset each time the system is started.
3. May be set automatically at start-up on ND-100 if this option has been included.

**Example:**

@DATCL

09. 15. 30      21 DECEMBER 1980

**@DEFAULT-SUBSYSTEM-DISABLE****Function:**

Disable execution of command string defined by @DEFINE-DEFAULT-SUBSYSTEM.

**Format:**

@DEFAULT-SUBSYSTEM-DISABLE

**Parameters**

<terminal number>

**Rules:**

1. Permitted for all users.
2. Only user SYSTEM can disable terminals other than their own.



## @ DEFAULT-SUBSYSTEM-ENABLE

**Function:**

Enable execution of command string defined by @DEFINE-DEFAULT-SUBSYSTEM

**Format:**

@DEFAULT-SUBSYSTEM-ENABLE

**Parameters**

<terminal number>

**Rules:**

1. Permitted for all users.
2. Only user SYSTEM can enable terminals other than their own.

## @ DEFINE-DEFAULT-SUBSYSTEM

**Function:**

This command is used to define a (global) command string, normally a recover command, to enter a user program immediately after LOGIN, ie. before "@" and after "OK". Although the command string is global it can be enabled and disabled for each terminal individually.

**Format:**

@DEFINE-DEFAULT-SUBSYSTEM

**Parameters:**

<subsystem name>

**Rules:**

Permitted only for user SYSTEM.

## @DEFINE-ESCAPE-CHARACTER

See also related @DISABLE-ESCAPE-FUNCTION and  
@ENABLE-ESCAPE-FUNCTION.

### Function:

Define the "user break" key for a specified terminal.

### Format:

@DEFINE-ESCAPE-CHARACTER <logical device no.>,<character value>

### Parameters:

<logical device no.>

must be a terminal (DEC, DEF = own terminal).

<character value>

ASCII value of new character to be the break character  
(OCT, DEF = 33 escape).

### Rules:

1. Permitted only for user SYSTEM.
2. Valid until the next time SINTRAN is started.

### Example:

@DEFINE-ESCAPE-CHARACTER 9,30

The escape character for terminal 9 is changed to 30 (cancel).

## @DEFINE-HISTOGRAM

See also @RT-PROGRAM-LOG and @DEFINE-SYSTEM-HISTOGRAM. See also related @START-HISTOGRAM, @STOP-HISTOGRAM and @PRINT-HISTOGRAM.

### Function:

Specify histogram parameters.

### Format:

@DEFINE-HISTOGRAM <program name>, <start address>, <interval>

### Parameters:

<program name>

the program being measured (DEF = background program of user's terminal).

<start address>

the virtual memory address where the measurements begin, ie. the starting address of the first interval (OCT, DEF = 0).

<interval>

the number of words in each of the 64 intervals. The intervals are contiguous (OCT).

### Rules:

1. Permitted for all users if <program name> is background program. Otherwise permitted only for users RT and SYSTEM.
2. The program is sampled at every basic time unit.

### Example:

@DEFINE-HISTOGRAM, 100, 10,

A histogram of the user's background program is to be produced. The range to be measured is 100<sub>8</sub> to 1077<sub>8</sub>, divided into 64<sub>10</sub> intervals of 10<sub>8</sub> words.

## @DEFINE-SPOOLING-CONDITIONS

See also related @DEFINE-SPOOLING-FILE-MESSAGE and @APPEND-SPOOLING-FILE.

### Function:

Define conditions under which spooling on a peripheral must operate.

### Format:

@DEFINE-SPOOLING-CONDITIONS <peripheral file name>,<printing name of spooling files?>,<stop and wait for start-print before printing file ?>,<no. of lines per page>

### Parameters:

<peripheral file name>

the spooling device.

<printing ... ?>

YES = file names will be printed on the error device when printing starts.

Otherwise, NO (DEF = NO).

<stop ... ?>

YES = automatic @STOP-PRINT between every printing of a file (see @DEFINE-SPOOLING-FILE-MESSAGE).

Otherwise, NO (can be YES only if <printing ... ?> = YES, DEF = NO).

<no. of lines per page>

number of lines per page as used by the commands @FORWARD-SPACE-PRINT and @BACKSPACE-PRINT (DEC, DEF = 68).

### Rules:

Permitted only for user SYSTEM.

### Example:

@DEFINE-SPOOLING-CONDITIONS LINE-PRINTER,NO,NO,

Spooling conditions are defined on the peripheral file LINE-PRINTER. Files are output without interruption.

## @DEFINE-SPOOLING-FILE-MESSAGE

See also related @DEFINE-SPOOLING-CONDITIONS.

### Function:

Define a text to be written on the error device whenever the terminal user's file is printed on the spooling device.

### Format:

@DEFINE-SPOOLING-FILE-MESSAGE <text>', <printing message independent of spooling conditions?>

### Parameters:

<text>

text to be written when printing starts. No text is specified by a ' (\$ is not converted to carriage return/line feed as in some other commands).

<printing message independent of spooling conditions?>

= YES, write text unconditionally.

= NO, write text only if requested in @DEFINE-SPOOLING-CONDITIONS (DEF = NO).

### Rules:

Permitted for all users.

### Example:

@DEFINE-SPOOLING-CONDITIONS OUTPUT FOR GUEST', YES

The message "OUTPUT FOR GUEST" is printed on the system error device whenever a file belonging to the owner user is output on the spooling device.

## @DEFINE-SYSTEM-HISTOGRAM

See also @DEFINE-HISTOGRAM and @RT-PROGRAM-LOG.

### Function:

Specify parameters for measuring the CPU time spent in different parts of the memory for a particular interrupt level.

### Format:

@DEFINE-SYSTEM-HISTOGRAM <level> , <start address> , <interval>

### Parameters:

<level>

any interrupt level less than 13 (DEC).

<start address> (OCT)

<interval>

see @DEFINE-HISTOGRAM (OCT).

### Rules:

1. Permitted only for user SYSTEM.
2. Sampling interval is basic time unit.
3. If used in conjunction with @RT-PROGRAM-LOG a distribution of interrupt levels 0 - 10 and above is given.

### Example:

@DEFINE-SYSTEM-HISTOGRAM 11, 60000, 100

A histogram for interrupt level 11 is produced. The range measured is 60000<sub>8</sub> to 67777<sub>8</sub>, divided into 64<sub>10</sub> intervals of 100<sub>8</sub> words.

## @DEFINE-TERMINATION-HANDLING

See also @NEXT-TERMINATION-COMMAND, @ENABLE-TERMINATION-HANDLING and EDTRM (MON 206).

### Function:

Define termination handling for background or RT-programs in case of termination with escape, MON 65 or fatal errors such as executing privileged instructions etc.

### Format:

@DEFINE-TERMINATION-HANDLING <RT or background>

### Rules:

1. In background a (global) set of commands is executed when interactive or mode (not batch) jobs are terminated abnormally.
2. In background the command is used like @INITIAL-COMMAND. The command buffer pointer is reset and the first command can be defined. If more than one command is to be defined, use @NEXT-TERMINATION-COMMAND.
3. In RT-termination the name of a user-defined RT-program should be given as a parameter. If enabled, this program will be started whenever a user RT-program terminates.
4. @MODE cannot be defined as a termination command and if a program is started (by a recover command) this will be the last command executed and termination will not be reenabled automatically.
5. Permitted only for user SYSTEM.

## @DELETE-BATCH-QUEUE-ENTRY

See also @ABORT-JOB. See also related @LIST-BATCH-QUEUE.

### Function:

Delete a job waiting in the batch queue.

### Format:

@DELETE-BATCH-QUEUE-ENTRY <batch no.>,<input file>,<output file>

### Parameters:

<batch no.>

batch processor number as returned from @BATCH (DEC).

<input file>

the file to be deleted.

<output file>

associated output file in batch queue.

### Rules:

1. Permitted only for user SYSTEM and the owner of the <input file>.
2. Valid only if second and third parameters match with a queue entry. These must be spelled the same way as in @APPEND-BATCH.
3. The currently active batch job is not in the queue and must be deleted by @ABORT-JOB.
4. If more than one entry matches, only the first one is deleted.
5. @LIST-BATCH-QUEUE may be used to find the correct spelling of the second and third parameters.

### Example:

@DELETE-BATCH-QUEUE-ENTRY 1, JOB-1, L-P

The batch job file JOB-1 is deleted from the queue to batch processor 1 (if matched). L-P was the output device in the corresponding @APPEND-BATCH command.



## @DELETE-FILE

See also @DELETE-USERS-FILES and MDLFI (MON 54).

**Function:**

Delete a file and release its pages.

**Format:**

@DELETE-FILE <file name>

**Parameters:**

<file name>

an unambiguous file name. The type must be specified (no DEF type). If a version is specified only that version is deleted (DEF = all versions).

**Rules:**

1. Permitted for all users with directory (D) access to the file.
2. If version  $v$  is deleted, the version number of  $v + 1$ ,  $v + 2$ , ... is decreased by 1.

**Example:**

File FILE-1:DATA;1 exists.

@DELETE-FILE FILE-1:D

The file is deleted.

## @DELETE-FRIEND

See also related @CREATE-FRIEND.

### Function:

Remove a user from the list of friends of the current user.

### Format:

@DELETE-FRIEND <user name>

### Parameters:

<user name>  
a friend of the current user.

### Rules:

1. Permitted for all users.
2. Note that <user name> may still have the current user as friend, ie. the friend relationship is not symmetric.

### Example:

@DELETE-FRIEND DESADE

The user DESADE is deleted as a friend of the current user.

## @DELETE-REENTRANT

See related @DUMP-REENTRANT.

### Function:

Delete a reentrant subsystem.

### Format:

@DELETE-REENTRANT <subsystem name>

### Parameters:

<subsystem name>  
name of the subsystem to be deleted.

### Rules:

Permitted only for user SYSTEM.

### Example:

@DELETE-REENTRANT NORD-PL

The subsystem named NORD-PL is deleted.

## @DELETE-REMOTE-QUEUE-ENTRY

See related @APPEND-REMOTE.

### Function:

Remove a remote batch input file from the remote batch queue to a remote computer. (The file is removed before it is sent to the remote computer.)

### Format:

@DELETE-REMOTE-QUEUE-ENTRY <remote computer> , <queue entry>

### Parameters:

<remote computer>

see @APPEND-REMOTE.

<queue entry>

a file name previously given in the @APPEND-REMOTE command (parameter must match exactly the remote batch queue entry and the user name must be given).

### Rules:

1. Permitted only for user SYSTEM and the owner of the file.
2. If the queue contains two or more equivalent entries, only the first one will be removed.

### Example:

@DELETE-REMOTE-QUEUE-ENTRY UNIVAC, (GUEST) JOB-1:SYMB

The batch-input-file (GUEST) JOB-1:SYMB is deleted from the batch queue for the host computer UNIVAC.

## @DELETE-SPOOLING-FILE

See also @REMOVE-FROM-SPOOLING-QUEUE.

### Function:

A file is removed from the spooling queue of a spooling device. If it is a spooling file, its pages are released and returned to the pool of free spooling pages and the file is marked as unused.

### Format:

@DELETE-SPOOLING-FILE <peripheral file name>,<file name>

### Parameters:

<peripheral file name>  
the spooling device.  
<file name>  
the spooling queue file to be deleted.

### Rules:

1. Permitted only for user SYSTEM and the user who appended the file.
2. Only the first name matching <file name> is removed.
3. If the file is not a spooling file the command is equivalent to @REMOVE-FROM-SPOOLING-QUEUE.

### Example:

@DELETE-SPOOLING-FILE LINE-PRINTER, LINE-PRINTER::10

If the file LINE-PRINTER::10 is in the spooling queue to LINE-PRINTER, it is deleted from the queue; if it is a spooling file, its pages are removed and it exists as an entry in the file system with no pages.

## @DELETE-USER

See also related @CREATE-USER, @GIVE-USER-SPACE and @TAKE-USER-SPACE.

### Function:

Remove a user from a directory.

### Format:

@DELETE-USER [<directory name>:]<user name>

### Parameters:

[<directory name>:]<user name>  
user to be removed on the particular directory (DEF directory = main directory).

### Rules:

1. Permitted only for user SYSTEM when disk is specified, for all users when floppy disk.
2. Valid only if the user has no files.

### Example:

@DELETE-USER USER-ONE

The user USER-ONE is deleted from their main directory.

## @DELETE-USERS-FILES

See also @DELETE-FILE.

### Function:

Delete one or more files for a user and release their pages.

### Format:

@DELETE-USERS-FILES <file name>[, <manual check?>]

### Parameters:

<file name>

delete all files matching this name (DEF = delete all files in default directory for the terminal user).

<manual check?>

YES = the name of every file matching <file name> is written on the terminal. The user specifies YES if it is to be deleted, otherwise, NO or return.

NO = delete all files matching <file name> immediately.  
(DEF = YES).

### Rules:

1. Permitted for all users with directory access (D) to the files.
2. If current user is SYSTEM, the <manual check?> parameter is not requested as manual check is compulsory.

### Example:

@DELETE-USER-FILES F-1, YES

FILE 5: (BIG-PACK:GUEST) F-10:DATA; 1? N

FILE 7: (BIG-PACK:GUEST) F-1:DATA; 1? Y

@

The file F-1:DATA is deleted but match was also found on the file F-10:DATA, which was not deleted.

## @DEVICE-FUNCTION

See also @LIST-DEVICE-FUNCTIONS and MAGTP (MON 144).

### Function:

Perform operations on magnetic tapes, cassette tapes, Versatec printer/plotter and floppy disk.

### Format:

@DEVICE-FUNCTION <peripheral file name> , <function> [, <optional parameter 1> [, <optional parameter 2> ]]

### Parameters:

<peripheral file name>

name of the peripheral file associated with the device.

<function>

see below.

<optional parameter 1>

see below.

<optional parameter 2>

see below.

### Rules:

Permitted for all users.

## @DEVICE-FUNCTION ADVANCE-RECORDS

### Purpose:

Advance over the specified number of records on a mag tape or cassette tape. On a floppy disk the disk address is incremented by the specified number of records.

### Format:

ADVANCE-RECORDS, <number of records>

<number of records> number of records or amount by which disk address is to be incremented.

### Rules:

Allowed on mag tape, cassette tape and floppy disk.

## @ DEVICE-FUNCTION ADVANCE-TO-EOF

**Purpose:**

Advance from the current position on a device over the specified number of end-of-file (EOF) marks. A tape will be positioned immediately after the last EOF mark passed. A floppy disk will have its address incremented to the address immediately after the last EOF mark passed.

**Format:**

ADVANCE-TO-EOF, <number of EOF marks>

**Rules:**

Allowed on magnetic tape, cassette tape and floppy disk.

## @ DEVICE-FUNCTION BACKSPACE-RECORDS

**Purpose:**

On mag tape or cassette move the tape backwards over the specified number of records. On floppy disk decrement the disk address by the specified number of records.

**Format:**

BACKSPACE-RECORDS, <number of records>

<number of records> number of records to backspace over or amount by which disk address is to be decremented.

**Rules:**

Allowed on mag tape, cassette tape and floppy disk.

## @ DEVICE-FUNCTION CLEAR-DEVICE

**Purpose:**

Abandon current operation of specified device. The device buffer on a Versatec is cleared.

**Format:**

CLEAR-DEVICE

**Rules:**

1. Allowed on mag tape, cassette, Versatec and floppy disk.
2. On mag tape clears all the units attached to one controller.

## @ DEVICE-FUNCTION CLEAR-SELECTED-UNIT

**Purpose:**

Clear the specified unit on the device.

**Format:**

CLEAR-SELECTED-UNIT, <unit>  
<unit>    device unit

**Rules:**

Allowed only on STC mag tapes.

## @ DEVICE-FUNCTION DUMP-BOOTSTRAP

**Purpose:**

Dump a bootstrap onto page 0 (the first page) of a floppy. The floppy monitor can then be loaded into a machine (not running SINTRAN) by pressing MASTER CLEAR and typing 1560&.

**Format:**

DUMP-BOOTSTRAP, <file name>  
<file name>    name of the :BPUN file containing the bootstrap. This will normally be FLOPPY-MONITOR:BPUN or a carefully prepared user monitor.

**Rules:**

Allowed only on floppy disk.

## @ DEVICE-FUNCTION ERASE-TAPE

**Purpose:**

Remove all recorded data from a cassette tape.

**Format:**

ERASE-TAPE

**Rules:**

Allowed only on cassette tape.



## @ DEVICE-FUNCTION FORMAT-FLOPPY

**Purpose:**

Write new address information onto the floppy in the currently selected format.

**Format:**

FORMAT-FLOPPY

**Rules:**

Allowed on floppy disk.

## @ DEVICE-FUNCTION GET-CURRENT-DISK-ADDRESS

**Purpose:**

Return the current address on the floppy disk. This is the address from which data will be written/read by the functions WRITE/READ-RECORD.

**Format:**

GET-CURRENT-DISK-ADDRESS

**Rules:**

Allowed on floppy disk.

## @ DEVICE-FUNCTION GIVE-FORM-FEED

**Purpose:**

Give form feed on a Versatec. On fan-fold paper this advances the paper to the top of the form; on roll paper it advances the paper 7cm.

**Format:**

GIVE-FORM-FEED, <number of form feeds>

<number of form feeds> number of form feeds, or 7cm advances, required.

**Rules:**

Allowed on Versatec.

## @ DEVICE-FUNCTION LOCK-CASSETTE

### **Purpose:**

The read/write heads of the cassette transport move onto the cassette tape ready for read/write. The cassette cannot be removed from the drive whilst the heads are locked.

### **Format:**

LOCK-CASSETTE

### **Rules:**

Can only be performed on cassette tape.

## @ DEVICE-FUNCTION READ-BYTE-RECORD

### **Purpose:**

Read specified number of bytes from peripheral to given address on user's background segment.

### **Format:**

READ-BYTE-RECORD, < destination address > , < number of bytes >  
 < destination address > address to which the data will be read (OCT).  
 < number of bytes > number of bytes to transfer (OCT).

### **Rules:**

Allowed on mag tape.

## @ DEVICE-FUNCTION READ-DELETED-RECORD

### **Purpose:**

Read a record even if it has been flagged as deleted.

### **Format:**

READ-DELETED-RECORD, < address > , < number of words >  
 < address > destination of words read to the user's background segment.  
 < number of words > number of words to transfer.

### **Rules:**

Allowed on floppy disk.

## @ DEVICE-FUNCTION READ-FORMAT

**Purpose:**

Read the type of address information (format) on a floppy disk.

**Format:**

READ-FORMAT

**Rules:**

1. Allowed only on floppy disk.
2. The format returned is 0, 1 or 2. For explanation see SET-FLOPPY-FORMAT.

## @ DEVICE-FUNCTION READ-LAST-STATUS

**Purpose:**

Read the status saved by the driver for the last operation on the device unit. An IOX instruction is not executed.

**Format:**

READ-LAST-STATUS

**Rules:**

1. Allowed on mag tape, cassette tape, Versatec and floppy disk.
2. For explanation of status word returned see Appendix I.

## @ DEVICE-FUNCTION READ-ODD-NUMBER-OF-BYTES

**Purpose:**

Read a specified odd number of bytes from the specified file to the user's background segment.

**Format:**

READ-ODD-NUMBER-OF-BYTES, < address > , < number of bytes >

< address > address of destination of data on the user's background segment.

< number of bytes > number of bytes to transfer.

**Rules:**

Allowed only on mag tape.

## **@ DEVICE-FUNCTION READ-STATUS**

**Purpose:**

Read the hardware status register of the specified device. The status is read by means of an IOX instruction.

**Format:**

READ-STATUS

**Rules:**

1. Allowed on mag tape, cassette tape, Versatec and floppy disk.
2. For explanation of status word returned see Appendix I.

## **@ DEVICE-FUNCTION READ-TAPE-STATUS**

**Purpose:**

Read the status word for a mag tape.

**Format:**

READ-TAPE-STATUS

**Rules:**

Allowed on mag tape.

## **@ DEVICE-FUNCTION REVERSE-TO-EOF**

**Purpose:**

Move from the current position backwards over the specified number of EOF marks. The device will be positioned just before the last EOF mark.

**Format:**

REVERSE-TO-EOF, <number of EOF marks>

**Rules:**

Allowed on mag tape, cassette tape and Versatec.

## @ DEVICE-FUNCTION REWIND

### Purpose:

On mag tape or cassette tape rewinds the tape to the load point; on floppy disk the disk address is reset to zero.

### Format:

REWIND

### Rules:

Allowed on mag tape, cassette tape and floppy disk.

## @ DEVICE-FUNCTION SELECT DENSITY

### Purpose:

Select the density with which data is written onto a mag tape.

### Format:

SELECT-DENSITY, <density>

<density> = 0 : 1600 Bits per inch  
              1 : 6250 Bits per inch

### Rules:

1. Allowed on Tandberg/Pertec and STC mag tape.
2. On some drive units density can be selected by switches on the front panel.

## @ DEVICE-FUNCTION SELECT-PARITY-AND-DENSITY

### Purpose:

Select the density and parity with which data are written onto mag tape.

### Format:

SELECT-PARITY-AND-DENSITY, <density/parity>

<density/parity> = 0 : 800 Bits per inch, odd parity  
                      1 : 556 Bits per inch, odd parity  
                      2 : 200 Bits per inch, odd parity  
                      3 : 800 Bits per inch, even parity  
                      4 : 556 Bits per inch, even parity  
                      5 : 200 Bits per inch, even parity

Default value is 0.

### Rules:

1. Allowed on Tandberg/Pertec and STC mag tapes.
2. On some drive units density can be selected by switches on the front panel.

## @ DEVICE-FUNCTION SET-ALPHANUMERIC-MODE

**Purpose:**

Put the Versatec into print mode. In this mode it functions like a line printer.

**Format:**

SET-ALPHANUMERIC-MODE

**Rules:**

Allowed on Versatec.

## @ DEVICE-FUNCTION SET-FLOPPY-FORMAT

**Purpose:**

SET-FLOPPY-FORMAT, <format number>

<format number>	=	0	:	256	words/sector, 8 sectors/track
					(standard format used by Norsk Data)
		1	:	128	words/sector, 15 sectors/track
		2	:	64	words/sector, 26 sectors/track

**Rules:**

Allowed on floppy disk.

## @ DEVICE-FUNCTION SET-GRAPHIC-MODE

**Purpose:**

Put the Versatec into plot mode. In this mode it functions like a graph plotter.

**Format:**

SET-GRAPHIC-MODE

**Rules:**

Allowed only on Versatec printer plotter.

## **@ DEVICE-FUNCTION UNLOAD**

**Purpose:**

Unload a mag tape, ie. the vacuum is removed from the tape and the tape rewound onto its spool.

**Format:**

UNLOAD

**Rules:**

1. Allowed on mag tape.
2. The tape must be at the load point (with the load point light illuminated) for this command to have effect.

## **@ DEVICE-FUNCTION UNLOCK-AND-STOP**

**Purpose:**

Stop cassette and remove the read/write heads from the tape. The cassette can then be removed from the drive.

**Format:**

UNLOCK-AND-STOP

**Rules:**

Allowed only on cassette.

## @ DEVICE-FUNCTION WRITE-BYTE-RECORD

**Purpose:**

Write specified number of bytes to peripheral file from the user's background segment.

**Format:**

WRITE-BYTE-RECORD, <address>, <number of bytes>

<address> address from which the bytes will be read.

<number of bytes> number of bytes transferred to the peripheral file.

**Rules:**

Allowed on mag tape.

## @ DEVICE-FUNCTION WRITE-DELETED-RECORD

Write a record and then flag it as deleted.

**Format:**

WRITE-DELETED-RECORD, <address>, <number of words>

<address> address on the user's background segment from which data is read.

<number of words> number of words to transfer.

**Rules:**

Allowed only on floppy disk.

## @ DEVICE-FUNCTION WRITE-EOF

**Function:**

Write specified number of EOF marks on the specified device. On floppy disk the device address is incremented by one for each EOF mark written.

**Format:**

WRITE-EOF, <number of EOF marks>

<number of EOF marks> number of EOF marks to write.



## **@ DEVICE-FUNCTION WRITE-ERASE-GAP**

**Function:**

Write a 4-inch (approx 10cm) gap on a tape.

**Format:**

WRITE-ERASE-GAP

**Rules:**

Allowed on mag tape and cassette tape.

## **@ DEVICE-FUNCTION WRITE-RECORD**

**Purpose:**

Write the specified number of words to the specified device from the specified address on the user's background segment.

**Format:**

WRITE-RECORD, <address> , <number of words>

<address> address on the user's background segment from which the data will be taken (OCT).

<number of words> number of words to be written (OCT).

**Rules:**

Allowed on mag tape, cassette tape, Versatec and floppy disk.

## @ DIRECTORY-STATISTICS

See also related @LIST-DIRECTORIES-ENTERED.

### Function:

List the following for one or more entered directories:

1. device name and unit number
2. directory status (default, main)
3. unreserved, unused and total space on the device.

### Format:

@DIRECTORY-STATISTICS <directory name> , <output file>

### Parameters:

<directory name>

the directory/directories to be listed (DEF = all directories entered).

<output file>

destination of listing (DEF = TERMINAL).

### Rules:

1. Permitted for all users.
2. The command requires some time to respond when referring to large disks.

### Example:

@DIRECTORY-STATISTICS, , ,

```
DISC-75MB-1 UNIT 1 : PACK-TWO
(MAIN AND DEFAULT DIRECTORY)
455 PAGES UNRESERVED AND 3529 PAGES UNUSED OUT OF 36945 PAGES
```

```
DISC-2-75MB-1 UNIT 0 SUBUNIT 0 : PACK-ONE
(MAIN AND DEFAULT DIRECTORY)
12 PAGES UNRESERVED AND 5174 PAGES UNUSED OUT OF 36945 PAGES
Statistics for all the entered directories are listed on the terminal.
```

## @DISABLE-ESCAPE-FUNCTION

See also related DESCF (MON 71), @ENABLE-ESCAPE-FUNCTION, EESCF (MON 72) and @DEFINE-ESCAPE-CHARACTER.

### Function:

The escape character is ignored in user and command modes, ie. no "user break" is performed.

### Format:

@DISABLE-ESCAPE-FUNCTION <logical device no.>

### Parameters:

<logical device no.>

must be a terminal (OCT, DEF = own terminal).

### Rules:

1. Permitted for all users. Only user SYSTEM can specify <logical device no.> other than their own terminal.
2. The escape function is enabled when logging out.

### Example:

@DISABLE-ESCAPE-FUNCTION,

No "user break" is performed for the user's own terminal.

## @DISABLE-TERMINATION-HANDLING

See also EDTRM (MON 206) and @ENABLE-TERMINATION-HANDLING.

### Function:

Disable termination handling for background and RT-programs.

### Format:

@DISABLE-TERMINATION-HANDLING <RT or background>, <terminal no.>  
(DEF = own terminal).

### Rules:

Only user SYSTEM can disable terminals other than their own.

**@DMAC**

See also DMAC (MON 51).

**Function:**

Special version of the MAC assembler used for debugging RT programs.

**Rules:**

1. Permitted for users SYSTEM and RT.
2. Cannot be restarted by @CONTINUE.

**@DSCNT**

See also DSCNT (MON 107). See also related @CONCT and CONCT (MON 106).

**Function:**

All connections to an RT program made by @CONCT (or monitor call CONCT) are removed. If the program is in the time queue, it is removed. Periodic execution is prevented.

**Format:**

@DSCNT <program name>

**Parameters:**

<program name>

RT program name or RT description address (OCT, DEF = own terminal background program).

**Rules:**

1. Permitted only for users RT and SYSTEM.
2. Reserved resources are not released.

**Example:**

@DSCNT KLOKK

The RT program KLOKK is disconnected.

**@DUMP**

See also @DUMP-REENTRANT. See also related @MEMORY, @PLACE-BINARY, @RECOVER and @CONTINUE.

**Function:**

Save the contents of the users virtual memory on a file.

**Format:**

@DUMP <file name> , <start address> , <restart address>

**Parameters:**

<file name>

mass-storage file to receive the memory contents (DEF type = PROG).

<start address>

the address where the program is started after @RECOVER  
(OCT, DEF = 0).

<restart address>

the address where the program is restarted after @CONTINUE  
(OCT, DEF = 0).

**Rules:**

1. Permitted for all users.
2. The amount of memory to be saved can be specified by @MEMORY. If not the values from the last @LOAD-BINARY, @PLACE-BINARY or @RECOVER are used.
3. The contents can be retrieved later by @RECOVER.

**Example:**

@MEMORY 100, 7777

@DUMP US-PROG, 100, 101

@RECOVER US-PROG

Virtual memory area 100<sub>8</sub> to 7777<sub>8</sub> is dumped on the program file US-PROG:PROG. The program is then started at address 100<sub>8</sub> by @RECOVER.

**@DUMP-BIT-FILE**

See also related @CHANGE-BIT-FILE.

**Function:**

Dump, in octal, the directory bit file on an output file.

**Format:**

@DUMP-BIT-FILE <directory name>, <block no.>, <output file>

**Parameters:**

<directory name>

directory whose bit file is to be dumped.

<block no.>

see below (DEC, DEF = 0).

<output file>

destination of the dump (DEF = TERMINAL).

**Rules:**

1. Permitted only for user SYSTEM.
2. The bit file is divided into 20<sub>8</sub> word blocks.

**Example:**

@DUMP-BIT-FILE B-P,,,

The first block of the bit file on directory B-P is dumped on the terminal.

## @DUMP-DIRECTORY-ENTRY

See also related @CHANGE-DIRECTORY-ENTRY.

### Function:

Dump, in octal, the directory entry part of the master block on an output file.

### Format:

@DUMP-DIRECTORY-ENTRY <device name> [, <unit> ] [, <'F' or 'R'> ]  
[, <subunit> ], <output file>

### Parameters:

<device name>  
the name of the device on which the directory resides.  
[ <unit> ]  
unit number on which the directory resides (if > 1).  
[ <'F' or 'R'> ]  
fixed or removable.  
[ <subunit> ]  
see @CREATE-DIRECTORY.  
<output file>  
destination of the octal dump (DEF = TERMINAL).

### Rules:

Permitted only for user SYSTEM.

### Example:

@DUMP-DIRECTORY-ENTRY DISC-10MB-1, 1, F,,

Make octal dump of directory on 10Mb disk, unit 1, fixed pack, on TERMINAL.

**@DUMP-OBJECT-ENTRY**

See also related ROBJE (MON 41) and @CHANGE-OBJECT-ENTRY.

**Function:**

Dump, in octal, the object entry of a file on an output file.

**Format:**

@DUMP-OBJECT-ENTRY [ <directory name>:] <user name>,<object no.>,  
<output file>

**Parameters:**

[ <directory name>:] <user name>  
directory and user where the file is to be found.  
<object no.>  
the object number of the file which may be found from @LIST-FILE.  
(DEC:DEF = 0).  
<output file>  
destination of the octal dump (DEF = TERMINAL).

**Rules:**

Permitted only for user SYSTEM.

**Example:**

@DUMP-OBJECT-ENTRY GUEST, 5,,

The object entry of object number 5 from user GUEST in default directory is dumped, in octal, on TERMINAL.



## @ DUMP-PAGE

See also related @CHANGE-PAGE.

**Function:**

Dump, in octal, a 1K page from a directory onto an output file.

**Format:**

@DUMP-PAGE <directory name> , <page address> , <output file>

**Parameters:**

<directory name>

directory which contains the page.

<page address>

physical page number (OCT, DEF = 0).

<output file>

destination of the octal dump (DEF = TERMINAL).

**Rules:**

Permitted only for user SYSTEM.

**Example:**

@DUMP-PAGE PACK-ONE, 100, ,

Page 100<sub>8</sub> in PACK-ONE is dumped, in octal, on the terminal.

## @DUMP-REENTRANT

See also related @DUMP, @DELETE-REENTRANT and @LIST-REENTRANT.

### Function:

Establish a subsystem as a reentrant subsystem. The subsystem is read from a file of type BPUN. The command then calls the RT loader and the subsystem is loaded to a segment. The subsystem name is inserted into a table which is searched by @RECOVER before the list of files for a PROG type file is searched.

### Format:

@DUMP-REENTRANT <subsystem name>, <start address>, <restart address>, <binary file>

### Parameters:

- <subsystem name>  
the name of the subsystem.
- <start address>  
the initial start address (OCT, DEF = 0).
- <restart address>  
the address where the program should be reentered by @CONTINUE (OCT, DEF = 0).
- <binary file>  
a file containing the subsystem in the format made by the )BPUN command of the MAC assembler (DEF type = BPUN).

### Rules:

1. Permitted only for user SYSTEM.
2. The command sets a protect flag ('ree-sub') on the segment.

### Example:

@COPY 'MAC:BPUN', (PRO-1:F-U)MAC:BPUN

@DUMP-REENTRANT MAC, -1, -3, MAC

The subsystem MAC is loaded to a segment and can be started by

@RECOVER MAC or @MAC.

**@DUMP-USER-ENTRY**

See also related RUSER (MON 44) and @CHANGE-USER-ENTRY.

**Function:**

Dump, in octal, a user entry onto an output file.

**Format:**

@DUMP-USER-ENTRY <directory name>,<user no.>,<output file>

**Parameters:**

<directory name>

<user no.>

number of the user found by @USER-STATISTICS (DEC, DEF = 0).

<output file>

destination of the octal dump (DEF = TERMINAL).

**Rules:**

Permitted only for user SYSTEM.

**Example:**

@DUMP-USER-ENTRY P-O, 4, ,

User entry number 4 on directory P-O is dumped on the terminal.

## @ENABLE-ESCAPE-FUNCTION

See also related @DISABLE-ESCAPE-FUNCTION, DESCF (MON 71) and @DEFINE-ESCAPE-CHARACTER.

**Function:**

Enable the "user break" function.

**Format:**

@ENABLE-ESCAPE-FUNCTION <logical device no.>

**Parameters:**

<logical device no.>

must be a terminal (OCT, DEF = the user's terminal).

**Rules:**

1. Permitted for all users. Only user SYSTEM can specify <logical device no.> other than their own.
2. This function is performed for the user's terminal when logging out.

**Example:**

@ENABLE-ESCAPE-FUNCTION,

The "user break" will be performed for the user's terminal whenever the escape key is pressed.

## @ENABLE-TERMINATION-HANDLING

See also EDTRM (MON 206) and @DISABLE-ESCAPE-HANDLING.

**Function:**

Enable termination handling for background and RT-programs.

**Format**

@ENABLE-TERMINATION-HANDLING <RT or background>,<terminal no.>  
(DEF = own terminal).

**Rules:**

Only user SYSTEM can enable escape handling for terminals other than their own.

**@ENTER**

See also related @APPEND-BATCH.

**Function:**

Identify the owner of a batch job; this must be the first command in a batch job.

**Format:**

@ENTER <user name>,<password>,<project password>,<maximum time>

**Parameters:**

<user name>

name of a user on a main directory.

<password>

the correct password for the user; it is not printed on the output listing (DEF = no password).

<project password>

<maximum time>

the maximum CPU time, in minutes, allowed for the total batch job. When time is up the job is aborted (DEC, DEF = 1).

**Rules:**

1. Permitted for all users.
2. If <password> is incorrect, the job is aborted.
3. If the job is run under @MODE, the @ENTER command is ignored.
4. A batch job does not need any special termination characters. A batch input file may contain several batch jobs.

**Example:**

@ENTER GUEST,,100,5

The command is located at the start of a batch job. The job is run for user GUEST with no user password and no project password. The job will terminate after 5 minutes CPU time.

## @ENTER-DIRECTORY

See also related @CREATE-DIRECTORY, @SET-DEFAULT-DIRECTORY, @SET-MAIN-DIRECTORY and @RELEASE-DIRECTORY.

### Function:

Enter a directory on a device. This must be done before files can be accessed.

### Format:

@ENTER-DIRECTORY <directory name>,<device name>[,<unit>]  
[,<'F' or 'R'>][,<subunit>]

### Parameters:

See @CREATE-DIRECTORY (DEF <directory name> is the name created on the device).

For <device name> see Appendix F.

### Rules:

1. Permitted for all users.
2. The directory is only entered if <directory name> matches or is an abbreviation of the name found on the device (see @CREATE-DIRECTORY).

### Example:

@ENTER-DIRECTORY PACK-ONE, DISC-10 MB-1, 1, F

A directory named PACK-ONE is entered on device DISC-10MB-1, unit 1, fixed pack.

**@ENTSG**

See also ENTSG (MON 157).

**Function:**

Initialize a page table with the specified segments and virtual address range and set the P-register.

**Format:**

@ENTSG <segment no.> , <page table> , <interrupt level> , <start address>

**Parameters:**

<segment no.>

the segment where the routine resides (OCT).

<page table>

the page table used for the routine (DEC:0-3, DEF = 0).

<interrupt level>

the interrupt level on which the routine is run (DEC). Legal values are interrupt levels 2, 5, 6, 7, 8 or 9. One of these levels must be specified as a dummy level even if the routine will be run on another level.

<start address>

start address of the routine (OCT).

**Rules:**

Permitted only for users RT and SYSTEM.

**Example:**

@ENTSG 42, 3, 9, 30000

The routine on segment 42 is entered to run as a direct task on page table 3, interrupt level 9. Location 30000<sub>8</sub> is the start address of the routine.

**@EXECUTE-IOX**

See also EXIOX (MON 31).

**Function:**

Execute an IOX instruction on a device.

**Format:**

@EXECUTE-IOX <value> , <device register address>

**Parameters:**

<value>

contents of A register before execution (OCT, DEF = 0).

<device register address>

bits 10 - 0 of the IOX instruction (OCT, DEF = 0).

**Rules:**

1. Permitted only for users RT and SYSTEM.
2. The contents of the A register are displayed after execution.
3. <device register address> must be listed in the internal SINTRAN III IOX table. Can be set by \*INSERT-IN-IOX-TABLE in @SINTRAN-SERVICE-PROGRAM.

**@EXPAND-FILE**

See also related @CREATE-FILE and @ALLOCATE-FILE.

**Function:**

Increase the length of a contiguous file.

**Format:**

@EXPAND-FILE <file name> , <no. of pages>

**Parameters:**

<file name>

the contiguous file to be expanded.

<no. of pages>

the additional number of pages (DEC).

**Rules:**

1. Permitted for all users.
2. File must be contiguous.

**Example:**

@EXPAND-FILE F-1, 9

The file F-1 is expanded by 9 pages.



## @ FILE-STATISTICS

See also @LIST-FILES and RMAX (MON 62). See also related @LIST-OPENED-FILES.

### Function:

List the following for the file(s):

1. File object number
2. File name
3. Type (indexed, contiguous, temporary, spooling or allocated file)
4. Access modes
5. Date of creation
6. Number of times opened
7. Last time opened for read and write
8. Size of file (in pages and bytes)

### Format:

@FILE-STATISTICS <file name> , <output file>

### Parameters:

<file name>

list all files matching this name. Can be abbreviated (DEF = all files in default directory for the current user).

<output file>

destination of the list (DEF = terminal).

### Rules:

Permitted for all users.

### Example:

@FILE-STATISTICS F-1, TER  
(statistics are listed)

**@FIX**

See also @FIXC and FIX (MON 115). See also related @UNFIX.

**Function:**

Fix a segment in memory, ie. it is not allowed to be swapped out. This command is useful for certain time-critical programs.

**Format:**

@FIX <segment no.>

**Parameters:**

<segment no.>  
(OCT).

**Rules:**

1. Permitted only for users RT and SYSTEM.
2. @UNFIX must be executed before the RT loader can clear the segment.
3. <segment no.> must be a nondemand segment.

**Example:**

@FIX 30

Segment number 30<sub>8</sub> is fixed in memory.

**@FIXC**

See also @FIX and FIXC (MON 160). See also related @UNFIX.

**Function:**

Fix a segment in a contiguous area of physical memory.

**Format:**

@FIXC <segment no.> , <first physical page no.>

**Parameters:**

<segment no.>  
(OCT).  
<first physical page no.>  
(OCT).

**Rules:**

1. Permitted only for users RT and SYSTEM.
2. @UNFIX must be executed before RT loader can clear the segment.
3. <segment no.> must be a nondemand segment.

**Example:**

@FIXC 40, 50

Segment number 40<sub>8</sub> is fixed in memory starting at physical page 50<sub>8</sub>, ie. address 120000<sub>8</sub>.

## @ FORWARD-SPACE-PRINT

See also related @BACKSPACE-PRINT, @STOP-PRINT and @START-PRINT.

**Function:**

Causes the spooling program to skip printing of the specified number of pages and lines in the current print file.

**Format:**

@FORWARD-SPACE-PRINT <peripheral file name> , <no. of pages> ,  
<no. of lines>

**Parameters:**

Similar to @BACKSPACE-PRINT.

**Rules:**

See @BACKSPACE-PRINT.

**Example:**

@STOP-PRINT

@FORWARD-SPACE-PRINT LINE-PRINTER, 2, ,

@START-PRINT

The spooling output for LINE-PRINTER is stopped and output is resumed two pages ahead.

## @ GET-ALTERNATIVE-RT-LOADER

See also related @MAKE-ALTERNATIVE-RT-LOADER.

### Function:

Load an alternative RT loader on segment file 0.

### Format:

@GET-ALTERNATIVE-RT-LOADER <file name>

### Parameters:

<file name>

a file where the RT loader was placed by @MAKE-ALTERNATIVE-SYSTEM  
(DEF type = DATA).

### Rules:

1. Permitted only for user SYSTEM.
2. To change to a new RT loader, the system must be started from MACM by means of the commands  
     )HENT  
     22!  
     Only the RT loader made for the current system can be used.

### Example:

@GET-ALTERNATIVE-RT-LOADER ALT-RT

The RT loader is loaded from the file ALT-RT:DATA to segment file 0.

## @GET-ALTERNATIVE-SYSTEM

See also related @MAKE-ALTERNATIVE-SYSTEM, @GET-ALTERNATIVE-RT-LOADER and @MAKE-ALTERNATIVE-RT-LOADER.

### Function:

Load SINTRAN from a specified area to the image area and start SINTRAN.

### Format:

@GET-ALTERNATIVE-SYSTEM <file name 1>,<file name 2>,<is RT loader in segment file no. 0?>[,<file name 3>],<is alt. "spooling program" in segment file no. 0?>,[<file name 4>]

### Parameters:

<file name 1>

specifies SINTRAN:DATA or a similar file containing the first part of the operating system. This file must consist of 63 contiguous pages on disk.

<file name 2>

specifies MACM-AREA:DATA or a similar file containing the rest of the SINTRAN III system except the RT loader, spooling system, NORDNET, SINTRAN-SERVICE-PROGRAM and MAIL system. It must be a contiguous file (of 64 pages) allocated immediately after <file name 1> (DEF type = DATA).

<is RT loader in segment file no. 0?>

YES = RT loader is taken from segment file number 0. <File name 1> and <file name 2> must be SINTRAN:DATA and MACM-AREA:DATA.

NO = a new RT loader is loaded from <file name 3> to segment file 0.

<file name 3>

name of the contiguous file (19 pages) where the RT loader resides. Specified only if previous parameter was NO.

<is alt. "spooling program" in segment file no. 0?>

YES = start the original system residing in SINTRAN:DATA, MACM-AREA:DATA and SEGFILO:DATA.

NO = another system on SEGFILO is loaded. Then the next parameter must be entered.

<file name 4>

name of the file where the spooling program, NORDNET, SINTRAN-SERVICE-PROGRAM and MAIL system reside. It must be a 24 page contiguous file. Specified only if the previous parameter was NO.

### Rules:

1. Permitted only for user SYSTEM.
2. The operating system is transferred to the memory image area (on SEGFILO) and the system is started. Since this is a "cold start" (similar to MACM — JHENT — 22!) all RT programs must be reloaded.
3. Subsequent restarts will use the alternative system. To return to the normal one use  
 @GET-ALTERNATIVE-SYSTEM SINTRAN,MACM-AREA,NO,  
 S-RTLOADER,YES  
 The RT loader must previously have been placed on the file  
 S-RTLOADER:DATA with the command  
 @MAKE-ALTERNATIVE-RT-LOADER S-RTLOADER:DATA.
4. All files in this command must reside in the same directory as the current system and on disk addresses below 77777.

## **@GET-ERROR-DEVICE**

See also related @SET-ERROR-DEVICE.

**Function:**

List the logical number of the device where error messages will be printed.

**Format:**

@GET-ERROR-DEVICE

**Parameters:**

None.

**Rules:**

Permitted only for user SYSTEM.

**Example:**

@GET-ERROR-DEVICE

ERROR DEVICE: 1

Error messages are printed on logical device number 1.

## @ GET-RT-NAME

See also GRTNA (MON 152).

**Function:**

Find the symbolic name of an RT program from the address of its RT description.

**Format:**

@GET-RT-NAME <address>

**Parameters:**

<address>

RT description address (OCT, DEF = background program of the user's terminal).

**Rules:**

1. Permitted only for users RT and SYSTEM.
2. Some system programs in SINTRAN have no symbolic name. They can be given names by the RT loader command DEFINE-PROGRAM.

**Example:**

@GET-RT-NAME , ,

BAK03

@

The user's terminal background program is BAK03.

## @ GET-TERMINAL-TYPE

See also MGTTY (MON 16). See also related @SET-TERMINAL-TYPE.

**Function:**

List the terminal type number of a terminal.

**Format:**

@GET-TERMINAL-TYPE <logical device no.>

**Parameters:**

<logical device no.>

must be a terminal (DEC, DEF = the user's terminal).

**Rules:**

1. Permitted for all users.
2. The ND standard for terminal types is shown in Appendix B.

**Example:**

@GET-TERMINAL-TYPE 9

TERMINAL TYPE: 1

@

Terminal type for terminal 9 is 1.

## @ GIVE-SPOOLING-PAGES

See also related @TAKE-SPOOLING-PAGES.

**Function:**

Increase the total number of pages that can be used by the spooling files.

**Format:**

@GIVE-SPOOLING-PAGES <no. of pages>

**Parameters:**

<no. of pages>  
additional no. of pages (DEC).

**Rules:**

1. Permitted only for user SYSTEM.
2. 500 pages are given to the spooling system when the system is generated.
3. The command does not guarantee that the disk space is available (see @START-SPOOLING, rule 2).

**Example:**

@GIVE-SPOOLING-PAGES 95

95 additional pages are added to the spooling pages.

## @ GIVE-USER-SPACE

See also related @TAKE-USER-SPACE.

**Function:**

Increase the total amount of reserved space for a user.

**Format:**

@GIVE-USER-SPACE [<directory name>:]<user name>,<no. of pages>

**Parameters:**

[.....:]<user name>  
directory and user to be given space (DEF directory = main directory).  
<no. of pages>  
additional number of pages (DEC).

**Rules:**

1. Permitted only for user SYSTEM when disk is specified, for all users when floppy disk.
2. Only the unreserved pages in the directory may be given to a user (use @DIRECTORY-STATISTICS).

**Example:**

@GIVE-USER-SPACE USER-ONE, 95

95 additional pages are given to the user USER-ONE in main directory.



## @GOTO-USER

See also related @RECOVER, @CONTINUE and @PLACE-BINARY.

**Function:**

Start background program at specific address.

**Format:**

@GOTO-USER <address>

**Parameters:**

<address>

(OCT, DEF = current P-register of the background program).

**Rules:**

1. Permitted for all users.
2. When a program is terminated by an "escape" or "break", all files are normally closed. To keep them open until @GOTO-USER is typed, @SET-PERMANENT-OPEN is used before the files are accessed for the first time.
3. After "escape" the values of the background RT-program registers may be examined by @STATUS.

**Example:**

```

@OPEN F-1:DATA, RX
FILE NUMBER IS: 101
@SET-PERMANENT-OPEN 101
@RECOVER MAIN
("escape")
@GOTO-USER

```

The program MAIN is started, stopped and finally resumed at the point of interruption. The file F-1:DATA is kept permanently opened so it is not closed at the "break".

**@HELP**

See also related \*HELP under @SINTRAN-SERVICE-PROGRAM,  
@LIST-DEVICE-FUNCTIONS and \*HELP under @MAIL.

**Function:**

List command names matching parameter.

**Format:**

@HELP <command> , <output file>

**Parameters:**

<command>

all commands matching <command> will be listed (DEF = all  
commands).

<output file>

destination of the list (DEF = TERMINAL).

**Rules:**

1. Permitted for all users.
2. If a command is a file system command, this is indicated. For all other  
commands, restrictions on its use (RT, SYSTEM only) are indicated.

**Example:**

```
@HELP EX,
RT: EXECUTE-IOX
FILSYS: EXPAND-FILE
@
```

**@ HOLD**

See also HOLD (MON 104).

**Function:**

Keep the terminal or batch processor waiting for a specified period.

**Format:**

@HOLD <no. of time units>, <time unit>

**Parameters:**

<no. of time units>

(DEC).

<time unit>

1 = basic time units (normally 20 ms).

2 = seconds

3 = minutes

4 = hours

**Rules:**

1. Permitted for all users.
2. No resources are released.
3. If the repeat bit in the RT-description is set there will be immediate return from HOLD.

**Example:**

@HOLD 5, 3

@

The second @ is prompted after 5 minutes.

## @INIT-ACCOUNTING

See related @START-ACCOUNTING and @STOP-ACCOUNTING.

### Function:

Initialize the accounting files and start accounting.

### Format:

@INIT-ACCOUNTING <desired>, <max.>, <background> [, <RT> ,  
<clear logged information>, <logging interval>] [, <ND-500>]

### Parameters:

<desired>  
number of account records before warning (DEC, DEF = 500).  
<maximum>  
maximum number of account records (DEC, DEF = 600).  
<background>  
are background programs to be accounted.  
<RT>  
are user RT programs to be accounted.  
<clear logged information>  
whether information already logged in the RT accounting table should be cleared.  
<logging interval>  
number of seconds between dumps of RT accounting table on file ACCOUNTS:DATA.

### Rules:

1. Permitted only for user SYSTEM.
2. When <desired> number of records is reached, the message "APPROACHING END OF ACCOUNTING FILE" is written on the terminal at every log out.
3. When <maximum> number of records is reached, the message "END OF ACCOUNTING FILE ENCOUNTERED" is written on the terminal at every log out. No further accounting is attempted until the next @INIT-ACCOUNTING or until ACCOUNTS:DATA is reset by the ACCOUNTING-SERVICE-PROGRAM.
4. If RT-accounting is not included in the system the question <RT> will not be asked. If the message "ACCRT NOT LOADED" is given after answering Y to <RT> the ACCRT program must be loaded onto segment 32 with the RT-loader (see System Supervisor manual, ND-30.003).
5. Only if the system contains an ND-500 will the question <ND-500> be asked.

### Example:

@INIT-ACCOUNTING 750, 850, Y, Y, Y, 600

The accounting system is initialized. A warning is given after 750 records and after 850 records no further logging takes place until the file is reset. Background and user RT-programs are accounted, information already in the RT-accounting table is cleared and dumps take place every 600 sec. This system does not contain an ND-500.

## @INITIAL-COMMAND

See also related @NEXT-INITIAL-COMMAND, @LIST-INITIAL-COMMAND and @RESTART-SYSTEM.

### Function:

Specify a command to be used at next restart from memory image, @RESTART-SYSTEM, or pressing MASTER CLEAR and LOAD buttons. The command must be @ENTER-DIRECTORY.

### Format:

@INITIAL-COMMAND <command string>

### Parameters:

<command string>  
the command is written without the @ sign.

### Rules:

1. Permitted only for user SYSTEM.
2. When @INITIAL-COMMAND has been issued, the @RTENTER is automatically executed at subsequent restarts.
3. If more commands are required @NEXT-INITIAL-COMMAND should be used.

### Example:

@INITIAL-COMMAND ENT-DIR P-ONE DIS-66MB-1

The command ENT-DIR P-ONE DIS-66MB-1 is executed at restart from memory image.

## @INITIALIZE-BACKGROUND-PROGRAMS

See also related @SET-AVAILABLE.

### Function:

Initiate the background terminal programs and batch processors.

### Format:

@INITIALIZE-BACKGROUND-PROGRAMS

### Parameters:

None.

### Rules:

1. Permitted only for user SYSTEM.
2. After a "cold start" (using the )HENT command in MACM) or after initial system load, the segment files must be defined (see \*DEFINE-SEGMENT-FILE under @SINTRAN-SERVICE-PROGRAM) and then @INITIALIZE-BACKGROUND-PROGRAMS is normally used.
3. Space is allocated for background programs on the segment file. @CHANGE-BACKGROUND-SEGMENT-SIZE must be used after @INITIALIZE-BACKGROUND-PROGRAMS.

## @INITIALIZE-ERROR-LOG

See also related @PRINT-ERROR-LOG.

### Function:

Initialize the segment on which error messages concerning hardware faults are recorded.

### Format:

@INITIALIZE-ERROR-LOG

### Parameters:

None.

### Rules:

1. Permitted only for user SYSTEM.
2. Error messages are collected on a segment for later printout. Messages concerning particular programs are not collected.

## @INTV

See also INTV (MON 103) and RT (MON 100).

### Function:

Prepare an RT program for periodic execution.

### Format:

@INTV <program name> , <no. of time units> , <time unit>

### Parameters:

<program name>

RT program name or RT description address (OCT, DEF = own terminal background program).

<no. of time units>

interval between each time the program is put in the execution queue.

<time unit>

see @HOLD.

### Rules:

1. Permitted only for users RT and SYSTEM.
2. The program must be activated by @CONCT, @RT or MON 100. It is then put in the time queue. At the end of the interval it is transferred from the time queue to the execution queue and at the same time put back in the time queue. This is repeated periodically.
3. @DSCNT or @ABORT stop periodic execution.

### Example:

@INTV KLOKK, 30, 2

@RT KLOKK

The program KLOKK is put in the execution queue every 30 seconds, starting when the @RT command is issued.

**@IOSET**

See also @CLEAR-DEVICE and IOSET (MON 141).

**Function:**

Set control information for device.

**Format:**

@IOSET <logical device no.>, <input/output>, <program name>, <control code>

**Parameters:**

<logical device no.>

(OCT).

<input/output>

0 = input part.

1 = output part.

<program name>

must be zero.

<control code>

-1 = reset device.

For card reader:

-1 = clear buffer and set ASCII mode. All characters are converted to ASCII and trailing blanks are ignored.

0 = set ASCII mode.

1 = set binary mode. Subsequent INBT monitor calls will return a 12 bit column image (OCT).

For synchronous modem output:

-1: clear modem output buffer

0: start sending of the current block

1: set 377 bytes to send between data

2: set ASCII synchronous to send

3: set EBCDIC synchronous to send

4: synchronize on 26 received (ASCII)

5: synchronize on 62 received (EBCDIC)

6: synchronize on 226 received (ASCII)

7: set ASCII synchronous (226 to send between data)

**Rules:**

1. Permitted for all users.
2. Device must be reserved by the terminal program.

**Example:**

@IOSET 5, 1, 0, -1

Reset line printer.

## @LIST-BATCH-PROCESS

See also @LIST-BATCH-QUEUE and @BATCH.

### Function:

List the state of each batch processor in the system.

### Format:

@LIST-BATCH-PROCESS

### Parameters:

None.

### Rules:

1. Permitted for all users.
2. A batch processor can be idle, active or passive; if it is active, the name of the logged on user is listed.
3. The command has no effect on the batch processor.

### Example:

```
@LIST-BATCH-PROCESS
      1 IDLE,    NO USER LOGGED ON
      2 ACTIVE,  USER GUEST LOGGED ON
      3 PASSIVE
```

Three processors are defined. The second is running a job belonging to user GUEST.

## @LIST-BATCH-QUEUE

See also related @LIST-BATCH-PROCESS

### Function:

List the contents of a specific batch queue.

### Format:

@LIST-BATCH-QUEUE <batch no.>

### Parameters:

<batch no.>  
       batch processor number as returned from @BATCH or  
       @LIST-BATCH-PROCESS.

### Rules:

1. Permitted for all users.
2. The command has no effect on the batch processor.

### Example:

```
@LIST-BATCH-QUEUE 1
1 CARD-READER LINE-PRINTER
2 (USER-NAME) BATIN LINE-PRINTER
```

The queue contains an entry from the card reader and one from the file (USER-NAME) BATIN.



## @LIST-DEFAULT-SUBSYSTEM

**Function:**

List the defined command string (if any) to be executed after login.

**Format:**

@LIST-DEFAULT-SUBSYSTEM <output file>

**Parameters:**

<output file>  
(DEF = own terminal).

**Rules:**

Permitted for all users.

## @LIST-DEVICE

See also related @LIST-RT-DESCRIPTION.

**Function:**

List the name of the RT-program which has reserved the device and the name of any RT programs waiting for the device.

**Format:**

@LIST-DEVICE <logical device no.> , <input/output>

**Parameters:**

<logical device no.>  
(DEC).  
<input/output>  
0 = input part.  
1 = output part.

**Rules:**

Permitted for all users.

**Example:**

@LIST-DEVICE 1, 1

RESERVED BY PROGRAM: BAK01

The output part of device 1 is reserved by the program BAK01.

## @LIST-DEVICE-FUNCTIONS

See also related @DEVICE-FUNCTION.

**Function:**

List all function names available in @DEVICE-FUNCTION.

**Format:**

@LIST-DEVICE-FUNCTIONS <output file>

**Parameters:**

<output file>

destination of the list (DEF = TERMINAL).

**Rules:**

Permitted for all users.

## @LIST-DIRECTORIES-ENTERED

See also @DIRECTORY-STATISTICS. See also related @CREATE-DIRECTORY and @ENTER-DIRECTORY.

**Function:**

List the names of entered directories and where they are mounted.

**Format:**

@LIST-DIRECTORIES-ENTERED <directory name, <output file>

**Parameters:**

<directory name>

list all directories matching this name, which can be abbreviated (DEF = all directories entered).

<output file>

destination of the list (DEF = TERMINAL).

**Rules:**

Permitted for all users.

**Example:**

@LIST-DIRECTORIES-ENTERED

DIRECTORY NAME: BIG PACK

OUTPUT FILE: (DEF = own terminal)

DISC-75MB-1 UNIT0: BIG-PACK

@

The directory matching the name BIG-PACK is listed on the terminal.

## @LIST-EXECUTION-QUEUE

See also related @LIST-TIME-QUEUE.

### Function:

List all the programs in the execution queue. The list contains the name or address of the RT description.

### Format:

@LIST-EXECUTION-QUEUE

### Parameters:

None.

### Rules:

1. Permitted for all users.
2. Programs are listed in the order in which they occur in the queue, ie. in order of decreasing priority.

## @LIST-FILES

See also @FILE-STATISTICS. See also related @LIST-OPENED-FILES.

### Function:

For each file, list file object number and full name.

### Format:

@LIST-FILES <file name>,<output file>

### Parameters:

<file name>

list all files matching this name, which can be abbreviated, (DEF = all files in default directory for current user).

<output file>

destination of the list (DEF = TERMINAL).

### Rules:

Permitted for all users.

## @LIST-FRIENDS

See also related @LIST-USERS, @USER-STATISTICS and @CREATE-FRIEND.

### Function:

List names and access modes of users declared as friends to the current user.

### Format:

@LIST-FRIENDS <user name> , <output file>

### Parameters:

<user name>

list all friends matching this user name, which can be abbreviated (DEF = all friends).

<output file>

destination of the list (DEF = TERMINAL).

### Rules:

Permitted for all users.

## @LIST-INITIAL-COMMANDS

See also related @INITIAL-COMMAND and @NEXT-INITIAL-COMMAND.

### Function:

List all commands defined as initial commands.

### Format:

@LIST-INITIAL-COMMANDS <output file>

### Parameters:

<output file>

destination of the list (DEF = TERMINAL).

### Rules:

1. Permitted only for user SYSTEM.
2. If user RT has files on a directory other than the main directory, two "next initial commands" must be used, @ENTER-DIRECTORY to enter the directory with RT files, @SET-DEFAULT-DIRECTORY to set it as default.
3. See @NEXT-INITIAL-COMMAND.

### Example:

@LIST-INITIAL-COMMANDS,

E-DIR B-P, D-66-1, 0

SET-ER-DEV 1

@

Two initial commands are currently defined.

## @LIST-OPENED-FILES

See also @LIST-RTOPENED-FILES. See also related @LIST-FILES and @FILE-STATISTICS.

**Function:**

List file number (logical device number) and full name of all the terminal user's opened files.

**Format:**

@LIST-OPENED-FILES <output file>

**Parameters:**

<output file>

destination of the list (DEF = TERMINAL).

**Rules:**

Permitted for all users.

**Example:**

@LIST-OPENED-FILES,

FILE NUMBER 000100: (BIG-PACK: SCRATCH) SCRATCH08: DATA; 1

@

The name of file no. 100 (SCRATCH file) for the terminal in use is SCRATCH08:DATA.

## @LIST-REENTRANT

See also related @DUMP-REENTRANT.

**Function:**

List available reentrant subsystems established by @DUMP-REENTRANT.

**Format:**

@LIST-REENTRANT

**Parameters:**

None.

**Rules:**

Permitted for all users.

**@LIST-REMOTE-QUEUE**

See also related @APPEND-REMOTE.

**Function:**

List the contents of a remote batch queue.

**Format:**

@LIST-REMOTE-QUEUE <remote computer>

**Parameters:**

<remote computer>

see @APPEND-REMOTE.

**Rules:**

Permitted for all users.

**Example:**

@LIST-REMOTE-QUEUE IBM

1 (SYSTEM) CARD-READER

2 (USER-NAME) IBMJOB

The queue contains an entry from the card reader and one from the file (USER-NAME)IBMJOB.

**@LIST-RT-ACCOUNT****Function:**

List accounting data for RT-programs.

**Format:**

@LIST-RT-ACCOUNT

**Parameters:**

None.

**Rules:**

Permitted for users RT and SYSTEM.

**@LIST-RT-DESCRIPTION**

See also related @LIST-RT-PROGRAMS.

**Function:**

List the information in the RT description on the terminal.

1. Current state information (repeat, IO wait, etc.).
2. Ring number.
3. Priority.
4. When last started or time left until next start.
5. Length of interval when periodic.
6. Start address.
7. Initial segment numbers.
8. Register contents.
9. READY, PASSIVE or WAITING FOR: <address>.
10. Actual segments.
11. Background if a background RT-program.
12. RTOFF if set.
13. Reentrant segment number if one is in use.
14. Address of data fields of reserved resources.

**Format:**

@LIST-RT-DESCRIPTION <program>

**Parameters:**

<program>

symbolic name of the RT program or address of the RT description (OCT, DEF = own terminal background program).

**Rules:**

Permitted for all users.

**Example:**

@LIST-RT-DESCRIPTION

RT NAME: (DEF = own terminal background program)

RING: 2

PRIORITY: 48

LAST STARTED: 27 MINS 58 SECS

START ADDRESS: 76055, SEGMENTS: 101 3

P = 130623

X = 2

T = 132

A = 1

D = 2

L = 130727

S = 1

B = 74460

READY

ACTUAL SEGM: 101 3 BACKGROUND

RESERVED DATAFIELDS:

24576

24523

## @LIST-RTOPENED-FILES

See also @LIST-OPENED-FILES. See also related @RTOPEN-FILE.

**Function:**

List all files opened for RT programs.

**Format:**

@LIST-RTOPENED-FILES <output file>

**Parameters:**

<output file>

destination of the list (DEF = TERMINAL).

**Rules:**

Permitted only for users RT and SYSTEM.

**Example:**

See @LIST-OPENED-FILES.

## @LIST-RT-PROGRAMS

See also related @LIST-RT-DESCRIPTION.

**Function:**

List the current status of all RT programs known by name to the RT loader.  
Information given is:

1. Name.
2. RT description address.
3. Priority.
4. Status (or state).
5. P register.
6. Time left until next start when in time queue.
7. Length of interval when periodic.
8. Actual segments.
9. RTOFF if set.

**Format:**

@LIST-RT-PROGRAMS <output file>

**Parameters:**

<output file>

destination of the list (DEF = TERMINAL).

**Rules:**

Permitted only for users RT and SYSTEM.



## @LIST-SEGMENT

See also \*WRITE-SEGMENTS in @RT-LOADER (manual ND-60.051).

**Function:**

List the following for a segment:

1. First logical page (the first page on page table 1 is logical page 100<sub>a</sub>, the first page on page table 2 is logical page 200<sub>a</sub>, etc.).
2. Length in pages.
3. Segment file number.
4. Relative page number within the segment file.
5. Status information (permit protect, demand/nondemand, reentrant sub-system flag, etc.).

**Format:**

@LIST-SEGMENT <segment no.>

**Parameters:**

<segment no.>  
(OCT).

**Rules:**

Permitted for all users.

## @LIST-SPOOLING-FORM

See also related @SET-SPOOLING-FORM.

**Function:**

List the identification key as set by @SET-SPOOLING-FORM.

**Format:**

@LIST-SPOOLING-FORM <peripheral file name>

**Parameters:**

<peripheral file name>  
a spooling device.

**Rules:**

Permitted for all users.

**Example:**

@LIST-SPOOLING-FORM LINE-PRINTER  
FORM-1

@

Only files with user text = FORM-1 are to be printed.

## @LIST-SPOOLING-QUEUE

See also related @SPOOLING-PAGES-LEFT and @DEFINE-SPOOLING-FILE-MESSAGE.

### Function:

List the following information about the spooling queue:

1. For the file being printed and those remaining in the queue list:
  - a) Full name
  - b) User who appended the file
  - c) Copies left to be printed
  - d) Bytes still to be printed in the current file.
2. If defined, a user message to be written on the error device when emptying the file (see @DEFINE-SPOOLING-FILE-MESSAGE).

### Format:

@LIST-SPOOLING-QUEUE <peripheral file name,<output file>

### Parameters:

<peripheral file name>  
the spooling device.  
<output file>  
destination of the list (DEF = TERMINAL).

### Rules:

Permitted for all users.

## @LIST-TERMINATION-HANDLING

### Function:

List commands defined by @DEFINE-TERMINATION-HANDLING.

### Format:

@LIST-TERMINATION-HANDLING

### Parameters:

<RT or background>

### Rules:

Lists commands defined with @DEFINE-TERMINATION-HANDLING and @NEXT-TERMINATION-COMMAND in background, or defined RT-program name in RT-termination.

**@LIST-TIME-QUEUE**

See also related @LIST-EXECUTION-QUEUE.

**Function:**

List the programs in the time queue. The list contains the name or address of RT descriptions.

**Format:**

@LIST-TIME-QUEUE

**Parameters:**

None.

**Rules:**

1. Permitted for all users.
2. Programs are listed in the order in which they occur in the queue, ie. in order of decreasing priority.

**@LIST-TITLE****Function:**

List system identification.

**Format:**

@LIST-TITLE

**Parameters:**

None.

**Rules:**

Permitted for all users.

**Example:**

@LIST-TITLE

SINTRAN III-VSE VERSION E

@

## @LIST-USERS

See also @USER-STATISTICS. See also related @LIST-FRIENDS.

### Function:

List full name of all users matching specified name.

### Format:

@LIST-USERS [ <directory name>:] <user name> , <output file>

### Parameters:

[ <directory name>:] <user name>

list all users matching this name, which can be abbreviated (DEF = all users in all default directories).

<output file>

destination of the list (DEF = TERMINAL).

### Rules:

Permitted for all users.

## @LOAD-BINARY

See also @PLACE-BINARY, @RECOVER and @REMOTE-LOAD.

### Function:

Load a program in BPUN format from a file to the user's memory and start execution.

### Format:

@LOAD-BINARY <file name>

### Parameters:

<file name>

program file name (DEF type = BPUN).

### Rules:

1. Permitted for all users.
2. The file must start with a bootstrap loader.
3. The addresses used for loading and starting are found in the program file, written by a )BPUN command (@MAC, @MACF, etc.). (The BPUN format is described in the NORD-100 Functional Description ND-06.015, Section VII. 2.5.)
4. When loading, a checksum is calculated and compared to the one in the program file. If they do not match execution is not attempted and an error message is given.
5. No restart address is defined, so restart with @CONTINUE is not possible.

### Example:

@LOAD-BINARY PROG-1

The program in the file PROG-1:BPUN is loaded and executed.

**@LOCAL**

See also @REMOTE.

**Function:**

Disconnect a remote connection.

**Format:**

@LOCAL

**Parameters:**

None.

**Rules:**

1. Permitted for all users.
2. Does not result in logout on the remote computer.

**@LOGOUT**

See also @STOP-TERMINAL and @TIME-USED.

**Function:**

Finish a session at the terminal by logging out the user and releasing the terminal.

**Format:**

@LOGOUT

**Parameters:**

None.

**Rules:**

1. Permitted for all users.
2. The time and date are given. The time used is given if the accounting system is active.

**Example:**

@LOGOUT

15. 20. 54            9 JANUARY 1981  
-- EXIT --

## @LOOK-AT

See also @STATUS and @SET-MEMORY-CONTENTS.

### Function:

Examine and/or modify memory locations and registers.

### Format:

@LOOK-AT <space reference> [, <additional parameter> ]

### Parameters:

<space reference>

MEMORY = user's virtual memory space. Permitted for all users with 128 pages background segment.

ALT-MEMORY = users alternative 64K memory. Addresses are specified relative to the 64K boundary. Permitted for all users.

SEGMENT = an existing segment. <additional parameter> = segment number. Permitted only for users RT and SYSTEM.

RTCOMMON = common area for RT programs. Permitted only for users RT and SYSTEM.

IMAGE = memory of the resident part of SINTRAN III on mass storage. A modification is permanent until a )HENT command overwrites the image. Permitted only for user SYSTEM.

RESIDENT = resident memory (0 - 177777<sub>8</sub>). This includes the resident part of SINTRAN. A modification is permanent until changed by loading from the image area. Permitted only for user SYSTEM. Page Tables (PT) can be accessed at the following octal addresses:

Normal mode:

(SINTRAN III-VS)

PT0: 177400 - 177477

PT1: 177500 - 177577

PT2: 177600 - 177677

PT3: 177700 - 177777

Extended mode:

(SINTRAN III-VSE, VSE 500)

PT0: 177000 - 177177

PT0: 177200 - 177377

PT2: 177400 - 177577

PT3: 177600 - 177777

REGISTERS = all registers on all hardware interrupt levels can be examined but only those on levels 2, 6, 7, 8, 9 and 15 can be changed. The register is specified as an octal number indicating the level, followed by the register name. Permitted only for users RT and SYSTEM.

### Rules:

1. Permission depends on the <space reference> as shown above.
2. When the <space reference> has been checked for legality, the message READY appears.
3. To examine a location, the octal address is typed followed by a slash (/). The contents are printed in octal and can now be changed by typing an octal value followed by CR. The value can be negative (preceded by a minus sign). If only CR is typed, the contents are unchanged and the contents of the next location are printed.
4. If an asterisk (\*) is typed, the current address will be printed.
5. For REGISTERS the valid names are P, X, T, A, D, L, S and B. Register names can also be specified in @LOOK-AT MEM.
6. When an illegal character is typed, a question mark is printed. Illegal characters have no effect. The command is terminated by a dot (.) or @.

7. If locations on mass storage segments are changed, the pages are written out so that "patches" are made permanent. Locations changed in the user's virtual memory or the resident part of the operating system are temporary. They may be altered when loading a user program or reloading the system.
8. If <space reference> is SEGMENT or IMAGE the changes only take place when the command is correctly terminated.

**Example:****@LOOK-AT MEMORY**

READY:

1000/ 10 100

11 .

@

Location 1000<sub>8</sub> in users virtual memory is changed from 10<sub>8</sub> to 100<sub>8</sub> (11<sub>8</sub> is the contents of location 1001<sub>8</sub>).

**@LOOK-AT REGISTERS**

READY:

7X/ 10 25

.

-END

@

The X register on level 7 is changed from 10<sub>8</sub> to 25<sub>8</sub>.

**@MAIL**

See also related @LOGOUT.

**Function:**

Enter the mail system, which is used for sending messages to other users or broadcasts to all users.

**Format:**

@MAIL [ <output file> ]

**Parameters:**

<output file>

destination of the mail from the terminal user's mailbox. Only requested if the user has mail (DEF = TERMINAL).

**Rules:**

1. Permitted for all users but some subcommands are restricted as shown below.
2. Messages can be sent in two ways:
  - a) to a mailbox — the recipient is notified when logging in or out and collects mail by entering @MAIL.
  - b) as direct mail — the message is sent immediately.
3. A broadcast is mail to all users, through the mailbox or as direct mail. It can only be sent by user SYSTEM.
4. The mail system can only be used by one user at a time.

**Subcommands:**

1. For all users the following subcommands are available:
  - \*EXIT — exit from the mail system.
  - \*HELP — list all available subcommands.
  - \*SEND-DIRECT-MESSAGE <logical device no.> — type message terminated by CTRL/L. The message is sent to the terminal with this <logical device no.>.
  - \*SEND-MESSAGE <user name> — type message terminated by CTRL/L. The message is sent to the user's mailbox. \$ and ' are handled as for \*BROADCAST below.



2. For user SYSTEM the following additional subcommands are available:
  - \*BROADCAST — type message terminated by CTRL/L. It is put in the mailbox of all users. \$ is translated to CR, LF. Apostrophe (') is permitted but terminates the message on output to its destination.
  - \*DELETE-BROADCAST <broadcast index> — the message is removed from all mailboxes. <broadcast index> can be found by \*LIST-BROADCASTS.
  - \*DELETE-MESSAGE <message no.> — the message is removed from the mailbox. The number can be found by \*LIST-MESSAGES.
  - \*DIRECT-BROADCAST — type message terminated by CTRL/L. The message is sent immediately to all terminals. \$ and ' are handled as for \*BROADCAST.
  - \*INITIALIZE <max. no. of messages> — this command must be given by user SYSTEM before the mail system can be used. It can be used to reset the mail system. The mail is collected in the file (SYSTEM)MAILBOX:DATA. The maximum length of a message is 512 characters.
  - \*LIST-BROADCASTS <output file> — all broadcasts are listed with their broadcast number on the output file (DEF = TERMINAL).
  - \*LIST-MESSAGES <output file> — as above, but messages are listed.
  - \*RUN-MAIL-SYSTEM — restarts the mail system after SINTRAN start or after a \*STOP-MAIL-SYSTEM command. The contents of the mailbox file are retained.
  - \*STOP-MAIL-SYSTEM — the mail system is made unavailable; no mail is lost.

## @ MAKE-ALTERNATIVE-RT-LOADER

See also related @MAKE-ALTERNATIVE-SYSTEM,  
 @GET-ALTERNATIVE-RT-LOADER and @GET-ALTERNATIVE-SYSTEM.

### Function:

Save the RT loader before a @GET-ALTERNATIVE-SYSTEM command.

### Format:

@MAKE-ALTERNATIVE-RT-LOADER <file name>

### Parameters:

<file name>

a contiguous file where the RT loader is saved.

### Rules:

1. Permitted only for user SYSTEM.
2. The file must be placed in the same directory as the current system and on disk address below 7777<sub>8</sub>.

## @MAKE-ALTERNATIVE-SYSTEM

See also related @MAKE-ALTERNATIVE-RT-LOADER,  
@GET-ALTERNATIVE-SYSTEM and @GET-ALTERNATIVE-RT-LOADER.

### Function:

Save the current system as an alternative system.

### Format:

@MAKE-ALTERNATIVE-SYSTEM <file 1>, <file 2>, <file 3>, <file 4>,  
<file 5>, <is source RT loader in segment file no. 0?>[, <file 6>] <file 7>,  
<is source SPOOLING/NORD-NET/SERVICE/MAIL-program in segment file  
no. 0?>[, <file 8>]

### Parameters:

- <file 1>  
a contiguous file with 63 pages where the first part of "alternative system"  
is placed (DEF type = DATA).
- <file 2>  
a contiguous file with 64 pages, allocated immediately after <file 1>  
which contains the second part of "alternative system" (DEF type =  
DATA).
- <file 3>  
a contiguous file with 20 pages where the "alternative RT loader" is be  
placed (DEF type = DATA).
- <file 4>  
a contiguous file where the first part of the "alternative system" is located  
(DEF type = DATA).
- <file 5>  
a contiguous file with 64 pages allocated immediately after <file 2> which  
contains the second part of the "alternative system" (DEF type = DATA).
- <is source RT loader in segment file no. 0?>  
YES = the current system is to be made alternative.  
Next parameter is not entered.  
NO = next parameter gives location of RT loader.
- <file 6>  
a contiguous file with 20 pages, containing the RT loader which is to be  
made alternative (DEF type = DATA).
- <file 7>  
a contiguous file with 26 pages where the "alternative" spooling program  
is placed. (Must be answered even if the system is without spooling.)
- <is source SPOOLING/NORDNET/SERVICE/MAIL-program in segment file  
no. 0?>  
YES = the current system is to be made alternative.  
NO = next parameter must be entered.
- <file 8>  
a contiguous file with 4 pages, where the spooling program resides which  
is to be made alternative.

### Rules:

1. Permitted only for user SYSTEM.
2. The system is loaded by @GET-ALTERNATIVE-SYSTEM. When returning to  
the normal system, the normal cold start procedure is followed by using  
MACM and )HENT. @GET-ALTERNATIVE-RT-LOADER is used to get the  
saved RT loader.
3. All the files must reside on the same directory as the current system and  
on disk addresses below 77777.

## @MEMORY

See also related @DUMP.

### Function:

Defines the area of virtual memory to be saved by the @DUMP command.

### Format:

@MEMORY <low address>,<high address>

### Parameters:

<low address>

<high address>

area to be saved, including the address limits (OCT, DEF = 0).

### Rules:

1. Permitted for all users.
2. The user's virtual memory space is unaffected.
3. The current boundary settings are reset by @LOAD-BINARY, @PLACE-BINARY and @RECOVER.

### Example:

@MEMORY 100,1777

The area to be dumped is 100 - 1777 inclusive.

## @MODE

See also @APPEND-BATCH.

### Function:

Enter SINTRAN commands from a file created by means of PED or TED and execute them.

### Format:

@MODE <input file>,<output file>

### Parameters:

<input file>

@MODE file containing commands.

<output file>

destination of the output normally made to TERMINAL. (If the command is being executed from the terminal (not mode or batch) default input and output files are the terminal; if it is being executed from a mode or batch job default input and output files are the currently open input and output files for that mode or batch job.)

**Rules:**

1. Permitted for all users.
2. Commands in a @MODE file should obey the following rules:
  - a) The first character of a command line in SINTRAN, MAIL or SINTRAN-SERVICE-PROGRAM commands must be @, which corresponds to the prompt character typed by the system in front of commands in direct mode. Commands to other subsystems such as the editor or loader must not be preceded by @ or any other character.
  - b) All command parameters must be specified on the same line as the command itself. The system cannot ask for missing parameters. The usual default parameters apply.
  - c) A @MODE file is terminated by escape (in TED/EDITOR this is typed as CTRL/V and "escape"; in PED this is typed as CTRL/O and "delete") or by end-of-file.
  - d) A @MODE command is permitted within a @MODE file or batch job.
  - e) Data input by the user can be interspersed with command lines in the same way as if they were typed from the terminal.
3. The MODE input and output files are assigned SINTRAN file numbers starting with the first available file number (normally 101<sub>8</sub>). These file numbers are not then available to user programs within the MODE file.
4. The file execution continues under the user currently logged on at the terminal from which the initial @MODE command was issued.
5. If the user program reads or writes data on device unit number 1 (file name: TERMINAL), such data will be taken from the <input file> and written on the <output file>.
6. If an error condition occurs within a @MODE job, a specific error message is written on the <output file>, the message \*\*\*BATCH JOB ABORTED\*\*\* appears on the terminal and execution is terminated.
7. If the end of the file is reached on the <input file>, control is returned to the terminal.
8. The execution may be interrupted by pressing the "escape" character but cannot then be resumed, as all opened files are closed. The last process initiated can be restarted by typing @CONTINUE (see @CONTINUE for exceptions).
9. The PERFORM subsystem makes it possible to include parameters in MODE files. (See Appendix H for further details.)

**Example:**

```

@MODE MF-1,
@DEL-FI F-1: DATA
@CRE-FI F-2, 0
@

```

A mode file is executed. It contains a @DELETE-FILE and a @CREATE-FILE command.

## @ MOVE-SPOOLING-QUEUE-ENTRY

See also related @APPEND-SPOOLING-FILE and @REMOVE-FROM-SPOOLING-QUEUE.

### Function:

Change the order of the files in the spooling queue.

### Format:

@MOVE-SPOOLING-QUEUE-ENTRY <peripheral file name>,<file name>,<insert or append?>,<before/after file name>

### Parameters:

<peripheral file name>

the spooling device.

<file name>

the file to be moved.

<insert or append>

insert = place <file name> in front of next parameter.

append = place <file name> after next parameter (can be abbreviated).

<before/after file name>

a file in the spooling queue referred to by the previous parameter.

### Rules:

Permitted only for user SYSTEM and the owner of the spooling queue file.

Only user SYSTEM may move entries forward in the queue.

### Example:

@MOVE-SPOOLING-QUEUE-ENTRY L-P, F-1, I, F-2

The file F-1 is moved in front of F-2 in the spooling queue.

## @NEXT-INITIAL-COMMAND

See also related @INITIAL-COMMAND and @LIST-INITIAL-COMMANDS.

**Function:**

Append a command to the set of commands executed at system start-up.

**Format:**

@NEXT-INITIAL-COMMAND <command string >

**Parameters:**

<command string>

the command is written without the @ sign.

**Rules:**

1. Permitted only for user SYSTEM.
2. The command buffer has room for a total of 256 characters.

**Example:**

@INITIAL-COMMAND ENT-DIR, P-ONE, DISC-66MB-1

@NEXT-INITIAL-COMMAND BATCH

@NEXT-INITIAL-COMMAND AP-BATCH, 1, ENT-FILE, L-P

A sequence of three commands is entered.

## @NORD-50

**Function:**

Enter a NORD-50 monitor. See the NORD-50 Monitor User's Guide and System Documentation (ND-60.076.02).

**Format:**

@NORD-50

**Parameters:**

None.

**Rules:**

1. Permitted for all users.
2. Applicable only to a system containing NORD-50s.

## @OPCOM

**Function:**

Enter the operator communication on the console of a NORD-100. It has the same function as pressing OPCOM on the panel.

**Format:**

@OPCOM

**Parameters:**

None.

**Rules:**

1. Permitted only for user SYSTEM.
2. The command must not be used in mode or batch jobs.
3. SINTRAN can be reentered from @OPCOM by pressing "escape".

## @OPEN-FILE

See also @CONNECT-FILE and @SCRATCH-OPEN. See also related @CLOSE-FILE, @SET-PERMANENTLY-OPENED, @RTOPEN-FILE and @RTCONNECT-FILE.

### Function:

Make a file available for access and return the file number used for accessing the file.

### Format:

@OPEN-FILE <file name>,<access type>

### Parameters:

<file name>

name of file to be opened (DEF type = SYMB).

<access type>

the intended access mode, a legal combination of:

R — read

W — write

X — random access

A — append

C — common access (contiguous file only)

The following combinations are legal:

R — sequential read

W — sequential write (access starts at beginning of mass-storage file)

RW — sequential read and write

RX — random read

WX — random read and write

WA — sequential write append (access starts at end of mass-storage file)

RC — random read with read and write access from other users allowed (contiguous files only)

WC — random read and write with read and write access from other users allowed (contiguous files only)

D — direct transfer (NORD-500 only)

DC — direct transfer with file closed (NORD-500 only)

### Rules:

1. Permitted for all users with sufficient <access type> to the file.
2. New file names and/or versions can be created when opening the file by enclosing <file name> in quotes. (DEF type = SYMB)
3. If file access is permitted, the message  
FILE NUMBER IS = n  
is printed. n is the <logical device no.> (OCT) to be used for later access.
4. An error message is produced if the file cannot be opened.
5. Files opened for sequential read and write occupy two device buffers. If all files are accessed sequentially, the maximum number of simultaneously opened files is 12. If all files are accessed randomly the number is 18.
6. Appendix C lists logical device numbers.

### Example:

@OPEN-FILE 'F-1:DATA',W

FILE NUMBER IS 000101

@

The indexed file F-1:DATA is created and opened for sequential write only. It is assigned file number 101<sub>8</sub>.



## @ OPERATOR

See also related @WAIT-FOR-OPERATOR and @RESTART-USER.

### Function:

Send message to error device.

### Format:

@ OPERATOR <text>

### Parameters:

<text>

any printable characters terminated by CR (DEF = no text).

### Rules:

1. Permitted for all users.
2. On the error device the following two lines are produced:  
 \*\*\*<time of day> TERMINAL <no.>  
 <text>

### Example:

@OPERATOR YOUR HOUSE IS ON FIRE!

On the error device the following message is produced:

\*\*\* 13. 25. 15 TERMINAL 52:

YOUR HOUSE IS ON FIRE!

## @PLACE-BINARY

See also @LOAD-BINARY. See also related @GOTO-USER.

**Function:**

Load a program in BPUN format but do not start it.

**Format:**

@PLACE-BINARY <file name>

**Parameters:**

<file name>

program file name (DEF type = BPUN).

**Rules:**

1. Permitted for all users.
2. The file must start with a bootstrap loader (@MAC, )BPUN format).
3. The program may be started by @GOTO-USER.
4. Programs in BRF format cannot be loaded.

**Example:**

@PLACE-BINARY USER-PROG

The program USER-PROG:BPUN is loaded to the user's memory but is not started.

## @PRINT-ERROR-LOG

See also related @INITIALIZE-ERROR-LOG.

**Function:**

Print the contents of the error log.

**Format:**

@PRINT-ERROR-LOG <output file>

**Parameters:**

<output file>

destination of the list (DEF = TERMINAL).

**Rules:**

1. Permitted only for user SYSTEM.
2. The log is not reset until next @INITIALIZE-ERROR-LOG.

**@PRINT-HISTOGRAM**

See also related @DEFINE-HISTOGRAM, @DEFINE-SYSTEM-HISTOGRAM, @START-HISTOGRAM and @STOP-HISTOGRAM.

**Function:**

Print the current histogram.

**Format:**

@PRINT-HISTOGRAM <output file>

**Parameters:**

<output file>

destination of the histogram (DEF = TERMINAL).

**Rules:**

1. Permitted for all users.
2. Valid only if the histogram is not printed before.
3. The sampling is stopped if this has not previously been done by @STOP-HISTOGRAM.
4. Producing the next histogram must start with @DEFINE-HISTOGRAM or @DEFINE-SYSTEM-HISTOGRAM.

**Example:**

@PRINT-HISTOGRAM, ,

	Percent	Samples
Outside:	12	10 out of 123
100 - 107:	00	0
110 - 117:	05	5
...	...	...
1070 - 1077:	00	0

(See example of @DEFINE-HISTOGRAM.)

**@PRIOR**

See also PRIOR (MON 110).

**Function:**

Set priority of an RT program. The priority is used to determine the sequence of queues (eg. execution queue, reservation or waiting queue for devices).

**Format:**

@PRIOR <program> , <priority>

**Parameters:**

<program>

RT program name or RT description address (OCT, DEF = user's terminal background program).

<priority>

(DEC: 0 - 255).

**Rules:**

1. Permitted only for users RT and SYSTEM.
2. Programs with priority = 0 will never start.

**Example:**

@PRIOR KLOKK, 80

The priority of RT program KLOKK is set to 80.

**@PRLS**

See also PRLS (MON 125). See also related @PRSRV.

**Function:**

Release a device from the program which reserved it.

**Format:**

@PRLS <logical device no.> , <input/output>

**Parameters:**

<logical device no.>

(DEC).

<input/output>

0 = input part of device

1 = output part of device

(DEF = 0).

**Rules:**

Permitted only for users RT and SYSTEM when resources are reserved.

**Example:**

@PRLS 18, 1

The output part of unit 18 is released from the RT program which reserved it.

**@PRSRV**

See also PRSRV (MON 124). See also related @PRLS.

**Function:**

Reserve a device for an RT program.

**Format:**

@PRSRV <logical device no.> , <input/output> , <program>

**Parameters:**

<logical device no.>

(DEC).

<input/output>

0 = input part of device

1 = output part of device

(DEF = 0).

<program>

RT program name or RT description address (OCT, DEF = user's terminal background program).

**Rules:**

1. Permitted only for users RT and SYSTEM.
2. Valid only if the unit is not reserved.

**Example:**

@PRSRV 18, 1, KLOKK

The output part of device 18<sub>s</sub> is reserved for the RT program KLOKK.

## @RECOVER

See also @LOAD-BINARY, @PLACE-BINARY and @CONTINUE. See also related @DUMP and @DUMP-REENTRANT.

### Function:

Invoke a subsystem on a reentrant segment, or load a program from a :PROG file to the user's virtual memory and start execution.

### Format:

@RECOVER <program name>

or

@ <program name>

### Parameters:

<program name>

reentrant subsystem name or program file name (if file, DEF type = PROG).

### Rules:

1. Permitted for all users.
2. If @RECOVER is omitted then <program name> must not coincide with a SINTRAN command (full or abbreviated). If it does, this command will be executed instead of @ <program name>. To force a search for a :PROG file rather than a command or reentrant subsystem, @RECOVER must be used.
3. Searching for the program name takes place as follows:
  - a) If no user name is specified, the reentrant subsystem names are searched (see @DUMP-REENTRANT).
  - b) If not found, <program name> is taken as the name of a :PROG file and the user default directory is searched.
  - c) If not found, user SYSTEM's file directory is searched.
  - d) If a user name is specified, only that user's file directory is searched.
4. The execution begins at the address specified in @DUMP, @DUMP-REENTRANT or \*DUMP (NRL subcommand).

### Example:

@RECOVER MAC or @MAC

The program MAC is loaded into memory and started.

**@REGENERATE-DIRECTORY**

See also @TEST-DIRECTORY.

**Function:**

Delete conflicting references to pages and reconstruct the bit file. This command must be used with care (see below).

**Format:**

@REGENERATE-DIRECTORY <directory name>

**Parameters:**

<directory name>  
an entered directory.

**Rules:**

1. Permitted only for user SYSTEM.
2. All files in the directory must be closed before the command is used. This includes scratch files, which are automatically closed when logging out and can also be closed by @CLOSE 100. User SYSTEM should log out all users and close his own scratch file before this command is used.
3. The command should be used with care. It may create "holes" in indexed files. This occurs when a reference to a page in an index block is set to zero because it conflicts with another index block or a contiguous file. When trying to read from this page the error message "No such page" is given. The user should take backup of all valuable files before this command is given.
4. If a page is included in an indexed and a contiguous file, it is deleted (0 in index block) from the indexed file.
5. The bit file is regenerated by this command.
6. This command may take several hours to complete on a large disk (up to 12 hours on a 10Mb disk and up to 3 days on a 288Mb disk). Stopping communication and RT-programs reduces the time required for regeneration.
7. It is advisable to use the FILE-SYSTEM INVESTIGATOR and back up the directory before this command is used.

**Example:**

@REGENERATE-DIRECTORY PACK-ONE

The directory PACK-ONE is regenerated.

## @RELEASE-DEVICE-UNIT

See also @RELEASE-FILE. See also related @RESERVE-DEVICE-UNIT and @RELEASE-DIRECTORY.

### Function:

Release a reserved device unit.

### Format:

@RELEASE-DEVICE-UNIT <device name>[, <unit>][, <'F' or 'R'>]

### Parameters:

<device name>

<unit>

<'F' or 'R'>

see @CREATE-DIRECTORY.

### Rules:

Permitted for all users.

### Example:

@RELEASE-DEVICE-UNIT FLOPPY-DISC-1,

Floppy disk 1 is now available to other users.

## @RELEASE-DIRECTORY

See also related @RELEASE-DEVICE-UNIT and @ENTER-DIRECTORY.

### Function:

Release an entered directory.

### Format:

@RELEASE-DIRECTORY <directory name>

### Parameters:

<directory name>

an entered directory.

### Rules:

1. Permitted for all users.
2. Valid only if no files in the directory are open and directory is not main directory and no non-main directories are entered.
3. After a directory is released it may be entered again, or another medium may be mounted on the unit and entered.
4. The command must be given before removing the medium.

### Example:

@RELEASE-DIRECTORY PACK-TWO

The directory PACK-TWO is released.



## @RELEASE-FILE

See also RELES (MON 123). See also related @PRLS, @RESERVE-FILE and @RELEASE-DEVICE-UNIT.

### Function:

Permit a peripheral file to be used from other terminals.

### Format:

@RELEASE-FILE <file name>

### Parameters:

<file name>  
must be a peripheral file.

### Rules:

Permitted for all users.

### Example:

@RELEASE-FILE LINE-PRINTER

The (peripheral) file LINE-PRINTER is released from the terminal user.

## @REMOTE

See also @START-COMMUNICATION and @STOP-COMMUNICATION. See also related @COMMUNICATION-STATUS and @LOCAL.

### Function:

Connect the terminal to a remote computer.

Send the local terminal type and ESCAPE character value to the remote computer.

If no remote connection exists for this terminal a free channel is found and the terminal is normally connected to the background processor of the remote computer. If a remote connection already exists, the terminal is connected to this channel. In this case, the @REMOTE command has the opposite function to pressing the "Rub-out" (or DEL) character.

### Format:

@REMOTE <line no.>

### Parameters:

<line no.>  
line number to remote computer (DEF = 1 if only one remote computer).

### Rules:

1. Permitted for all users.
2. The local terminal type will be sent to the remote computer.
3. Protocol version must be the same on all computers.

### Example:

@REMOTE 2

The terminal is connected to the remote computer on line 2.

## @REMOTE-LOAD

See also @LOAD-BINARY.

### Function:

Load a stand-alone program into a remote computer.

### Format:

@REMOTE-LOAD <load file> , <bootstrap address> , <line no.>

### Parameters:

<load file>

file name of program in :BPUN format.

<bootstrap address>

starting address of bootstrap in remote computer. It occupies approximately 400 words (DEF = 0).

<line no.>

line number to remote computer (DEF = 1 if only one remote computer).

### Rules:

1. Permitted for users RT and SYSTEM.
2. Valid only when communication is stopped on the line.
3. The bootstrap must reside on a system file (SYSTEM)REMOTE-BOOTSTRAP:BPUN, in the local computer. The file must be in :BPUN format. It is usually generated at system generation time.
4. The remote computer must be initiated by pressing MASTER CLEAR and typing <octal no.> & on the console terminal before doing a remote load from a local computer. <octal no.> is the hardware device number of the communication line. If automatic load is used, only MASTER CLEAR and LOAD is required.

### Example:

@REMOTE-LOAD REMFI 37400,1

The file called REMFI on the local computer will be loaded to the remote computer on line 1. The bootstrap will occupy the memory area 37400 - 37777 in the remote computer.

**@REMOTE-PASSWORD**

See also related @CHANGE-PASSWORD.

**Function:**

Set the password to be used upon remote open file from RT programs.

**Format:**

@REMOTE-PASSWORD <line no.>,<password>

**Parameters:**

<line no.>

line number to remote computer (DEF = 1, if only one remote computer).

<password>

(DEF = no password).

**Rules:**

1. Permitted only for user SYSTEM.
2. The password is used when a file is opened through <line no.> by an RT program. The file is only opened if the password of user RT in the remote computer is the same as this password.
3. See rules 3 and 4 of @CHANGE-PASSWORD.

**Example:**

@REMOTE-PASSWORD 3, DELTA

Files are only opened in the remote computer on line number 3 if user RT has password DELTA.

## @ REMOVE-FROM-SPOOLING-QUEUE

See also @DELETE-SPOOLING-FILE. See also related  
 @APPEND-SPOOLING-FILE, @MOVE-SPOOLING-QUEUE-ENTRY and  
 @LIST-SPOOLING-QUEUE.

### Function:

Remove a file from the spooling queue of a spooling device. The contents of the file are retained.

### Format:

@REMOVE-FROM-SPOOLING-QUEUE <peripheral file name>, <file name>

### Parameters:

<peripheral file name>  
     spooling device.  
 <file name>  
     the spooling queue file to be removed.

### Rules:

1. Permitted only for user SYSTEM and the user who appended the file to the queue.
2. Only the first name matching <file name> is removed.

### Example:

@REMOVE-FROM-SPOOLING-QUEUE LINE-PRINTER, F-1

A search is made for the file name F-1 in the spooling queue to LINE-PRINTER. If found it is removed from the queue.

## @RENAME-DIRECTORY

See also related @ENTER-DIRECTORY.

### Function:

Change the name of a directory.

### Format:

@RENAME-DIRECTORY <old directory name>, <new directory name>,  
<device name>[, <unit>][<'F' or 'R'>][, <subunit>]

### Parameters:

<old directory name>

(DEF = name of directory on medium of specified device).

<new directory name>

<device name>

see Appendix F.

<unit>

<'F' or 'R'>

<subunit>

see @CREATE-DIRECTORY.

### Rules:

1. Permitted only for user SYSTEM when disk is specified, for all users when floppy disk.
2. Valid only if the directory is not entered.
3. Valid only if the old and new directory names are not entered on any other device unit.

### Example:

@RENAME-DIRECTORY F-2, F-1, F-D-1

Change directory name on floppy disk controller 1 from F-1 to F-2.

**@RENAME-FILE**

See also related @CREATE-FILE.

**Function:**

Change file object name and/or type.

**Format:**

@RENAME-FILE <old file name>,[<new object name>],[:<new type>]

**Parameters:**

<old file name>  
     an existing file.  
 <new object name>  
     (DEF = no change).  
 <new type>  
     (DEF = no change).

**Rules:**

1. Permitted for all users with directory access to the file.
2. If version is specified in <old file name> only that version is renamed, otherwise all versions are renamed.
3. Directory name, user name and version number should not be specified in the new file name.

**Example:**

@RENAME-FILE (P-T:GUEST) F-1:SOU, :SYMB

The file (P-T:GUEST) F-1:SOU is changed to type :SYMB.

## @RENAME-USER

See also related @CREATE-USER.

**Function:**

Change the name of a user in a directory.

**Format:**

@RENAME-USER [`<directory name>` :]`<old user name>` ,`<new user name>`

**Parameters:**

`<directory name>`

directory where the user resides (DEF = main directory).

`<old user name>`

an existing user.

`<new user name>`

the new name.

**Rules:**

1. Permitted only for user SYSTEM when disk is specified, for all users when floppy disk.
2. Valid only if the user has closed all his files.
3. The user name is changed in one directory only.
4. The directory name should not be specified in the new user name.

**Example:**

@RENAME-USER GUEST, GJEST

The name of the user GUEST in main directory is changed to GJEST.

## @RESERVE-DEVICE-UNIT

See also related @RELEASE-DEVICE-UNIT and @RESERVE-FILE.

### Function:

Reserve device for special use. No directories can be entered until the device is released.

### Format:

@RESERVE-DEVICE-UNIT <device name> [, <unit> ] [, <'F' or 'R'> ]  
[ , <subunit> ]

### Parameters:

<device name>  
<unit>  
<'F' or 'R'>  
see @CREATE-DIRECTORY.

### Rules:

1. Permitted for all users.
2. The command is not executed if a directory is already entered on the unit.

### Example:

@RESERVE-DEVICE-UNIT FLOPPY-DISC-1,  
No directory can be entered on FLOPPY-DISC-1.

## @RESERVE-FILE

See also RESRV (MON 122). See also related @PRSRV, @RELEASE-FILE and @RESERVE-DEVICE-UNIT.

### Function:

Reserve a peripheral file for the exclusive use of the terminal user.

### Format:

@RESERVE-FILE <file name>

### Parameters:

<file name>  
must be a peripheral file.

### Rules:

1. Permitted for all users.
2. The reserved file is released on LOGOUT.
3. Not permitted for batch jobs (see @SCHEDULE).

### Example:

@RESERVE-FILE LINE-PRINTER  
The (peripheral) file LINE-PRINTER is reserved for the terminal user.



## @RESTART-PRINT

See also @START-PRINT.

### Function:

Restart the printing of the file currently being printed on the spooling device.

### Format:

@RESTART-PRINT <peripheral file name>

### Parameters:

<peripheral file name>  
spooling device.

### Rules:

1. Permitted only for user SYSTEM and the user who appended the file.
2. Valid only if the spooling program is started (@START-SPOOLING) and a file is being printed. The command is useful eg. when there has been a paper crash on the line printer.

### Example:

@RESTART-PRINT LINE-PRINTER

Printing of the current file on the peripheral LINE-PRINTER is restarted.

## @RESTART-SYSTEM

See also related @STOP-SYSTEM.

### Function:

Restart the system by simulating a STOP, MASTER CLEAR and LOAD.

### Format:

@RESTART-SYSTEM

### Parameters:

None.

### Rules:

1. Permitted only for user SYSTEM.
2. All RT programs and segments loaded by the RT loader are retained. All files should be closed and no directories other than the main directory should be entered.
3. The commands defined by @INITIAL-COMMAND and @NEXT-INITIAL-COMMAND are executed.
4. This command is optional and can be included at system generation time.
5. This command cannot be abbreviated.

## @RESTART-USER

See also related @OPERATOR and @WAIT-FOR-OPERATOR.

**Function:**

Restart the user on a terminal after he has used @WAIT-FOR-OPERATOR.

**Format:**

@RESTART-USER <logical device no.>

**Parameters:**

<logical device no.>  
must be a terminal (DEC).

**Rules:**

Permitted only for users RT and SYSTEM.

**Example:**

@RESTART-USER 9

\*\*\* 09.46.37 RESTARTED TERMINAL 9

Terminal 9 is restarted for the user.

**@RFILE**

See also RFILE (MON 117). See also related @WFILE, WFILE (MON 120) and @SET-BLOCK-SIZE.

**Function:**

Transfer one or more blocks from a file opened for random read into the user's memory.

**Format:**

@RFILE <file no.>, <memory address>, <block no.>, <no. of words>

**Parameters:**

<file no.>

logical device number from the @OPEN-FILE or @CONNECT-FILE (OCT).

<memory address>

starting address of memory destination (OCT, DEF = 0).

<block no.>

block number to be accessed (OCT, DEF = 0).

<no. of words>

length of the block in words (DEC, DEF = 256).

**Rules:**

1. Permitted for all users.
2. Can also be used on communication channels.
3. Default block size is 256<sub>10</sub>.
4. When using direct access mode, the number of words read/write must be a multiple of physical block (sector) size.

**Example:**

@RFILE 101, 400, 0, 1000

Read 1000<sub>8</sub> words from block 0 of file 101<sub>8</sub> to memory address 400<sub>8</sub>.

**@RT**

See also RT (MON 100). See also related @SET and @ABSET.

**Function:**

Put a program in the execution queue.

**Format:**

@RT <program>

**Parameters:**

<program>

RT program name or address of RT description (OCT, DEF = user's terminal background program).

**Rules:**

1. Permitted only for users RT and SYSTEM.
2. If the program is already in the execution queue the repeat bit will be set.
3. If the program is in HOLD or RTWT it is removed from that state. In the case of HOLD it is not removed from the time queue; the program is restarted or its repeat bit set when the HOLD time expires.

**Example:**

@RT KLOKK

The RT program KLOKK is put in the execution queue.

**@RT-LOADER****Function:**

Start RT loader. See manual ND-60.051.

**Rules:**

1. Permitted only for users RT and SYSTEM.
2. The RT loader can only be used by one user at a time.
3. Cannot be restarted by @CONTINUE.

## ① RT-PROGRAM-LOG

See also related @START-PROGRAM-LOG.

### Function:

Measure resource usage by logging a particular RT program and total system usage. Usage of one or two logical units can also be logged. (See "The Analysis of Response Time Problems", ND Bulletin, December 1980.)

### Format:

@RT-PROGRAM-LOG <program name>,<report interval>,<interrupts/sample>,<logical device no.>[,<input or output>][,<logical device no.>][,<input or output>],<output file>

### Parameters:

- <program name>
  - the name of the RT program to be measured or the address of its RT description (DEF = log only total system usage).
- <report interval>
  - the number of seconds between each report time.
- <interrupts/sample>
  - the number of output interrupts of the user's terminal between each sample.
- <logical device no.>
  - any existing logical device to be measured. Two devices can be measured at one time (DEC, DEF = no log).
- <input/output>
  - input or output of the logical device
  - 0 = input
  - 1 = output
- <output file>
  - destination of the report (DEF = TERMINAL).

### Rules:

1. Permitted only for users RT and SYSTEM.
2. Sampling is connected to the output interrupt response time of the terminal. If the terminal is of high speed, e.g. 9600 baud, a sample on each interrupt would give a considerable system load. The parameter <interrupts/sample> should specify a number which is reasonable in the particular circumstances.
3. The information returned is:
  - CPU time used
  - disk channel time for swapping
  - disk channel time for file access
  - the time the disk was reserved; this is the total of swap and file access time the logical unit(s) is(are) reserved when the operating system itself or an RT program is measured.
 For RT programs additional information is given about:
  - time when program was passive
  - time when program was waiting for I/O interrupts.
 Information is given as percentages. Eg. "20/45" returned as CPU load means that the RT program uses 20% of the total available CPU time, while the total system uses 45%. The rest of the time the CPU is idle.
4. To terminate the log, press "escape".
5. The command is not permitted from remote terminals.

**Example:**

```
@RT-PROGRAM-LOG  BAK02, 1, 1, ,
```

CPU	SWAP	FILES	DISK	PASSIVE	IO WAIT
00/ 95	00/ 00	00/ 00	00	94	04
41/ 98	00/ 00	13/ 13	13	00	60
44/ 97	00/ 00	03/ 03	03	00	57
63/100	00/ 00	00/ 00	00	00	24
24/100	00/ 00	00/ 00	00	00	10
65/100	00/ 00	00/ 00	00	00	33
63/100	00/ 00	01/ 01	01	00	33
04/100	00/ 00	00/ 00	00	00	00
63/100	00/ 00	01/ 01	01	00	33
65/100	00/ 00	00/ 00	00	00	33
65/100	00/ 00	01/ 01	01	00	32
05/100	00/ 00	00/ 00	00	00	00
10/100	00/ 00	01/ 01	01	00	04
03/100	00/ 00	00/ 00	00	00	00
50/100	00/ 00	03/ 03	03	00	28
04/100	00/ 00	00/ 00	00	00	00

The program BAK02 is logged every second. One sample is taken on every interrupt of terminal 2. The second row shows that CPU was utilized 98% of the time and BAK02 used 41% of that time. The disk was reserved 13% of the time, all for BAK02 file access. BAK02 was in IO-WAIT 60% of the time. The sum of CPU time, passive time and IO-WAIT is 101%, due to rounding inaccuracy.

## @RTCLOSE-FILE

See also @CLOSE-FILE. See also related @RTOPEN-FILE and @RTCONNECT-FILE.

### Function:

Close a file opened for RT programs.

### Format:

@RTCLOSE-FILE <file no.>

### Parameters:

<file no.>

> 0: close the file with the specified logical device no.

- = -1 or = -2: close all files. This applies to all RT programs.

### Rules:

Permitted only for users RT and SYSTEM.

### Example:

@RTCLOSE-FILE 101

The file 101<sub>a</sub> is closed for all RT programs.

## @RTCONNECT-FILE

See also @CONNECT-FILE and @RTOPEN-FILE. See also related @RTCLOSE-FILE.

### Function:

Open a file with a previously defined number. The file will be available for all RT programs.

### Format:

@RTCONNECT-FILE <file name>, <file no.>, <access type>

### Parameters:

<file name>

(DEF type = SYMB).

<file no.>

any free number (OCT 100 - 121).

<access type>

see @OPEN-FILE.

### Rules:

1. Permitted only for users RT and SYSTEM.
2. See rules 2 - 6 of @OPEN-FILE.
3. When accessing files in RT programs user RT is the user.
4. RT opened files cannot be set permanently open.

### Example:

@RTCONNECT-FILE F-1, 120, R

File F-1:SYMB is opened for read by all RT programs and given file number 120<sub>a</sub>.

N

**@RTENTER**

See also @INITIAL-COMMAND.

**Function:**

Enable RT programs to open, close, read and write to files.

**Format:**

@RTENTER

**Parameters:**

None.

**Rules:**

1. Permitted only for users RT and SYSTEM.
2. This command is also executed when @INITIAL-COMMAND is executed.
3. The command must always be given when starting SINTRAN III.

**@RTOFF**

See also RTOFF (MON 137). See also related @RTON.

**Function:**

Disable the starting of an RT-program until @RTON is given.

**Format:**

@RTOFF <program>

**Parameters:**

<program>

RT program name or RT description address (OCT, DEF = user's terminal background program).

**Rules:**

1. Permitted only for users RT and SYSTEM.
2. RT programs already running will not be stopped (see @ABORT).

**Example:**

@RTOFF KLOKK

The RT program KLOKK cannot be started until @RTON is given.



## @RTON

See also RTON (MON 136). See also related @RTOFF.

**Function:**

Enable starting of an RT program.

**Format:**

@RTON <program>

**Parameters:**

<program>

RT program name or RT description address (OCT, DEF = user's terminal background program).

**Rules:**

Permitted only for users RT and SYSTEM.

**Example:**

@RTON KLOKK

The RT program KLOKK can now be started.

## @RTOPEN-FILE

See also @OPEN-FILE, OPEN (MON 50) and @RT-CONNECT-FILE. See also related @CLOSE-FILE and @RTCLOSE-FILE.

**Function:**

Open a file for RT programs.

**Format:**

@RTOPEN-FILE <file name> , <access type>

**Parameters:**

<file name>

(DEF type = SYMB).

<access type>

see @OPEN-FILE.

**Rules:**

1. Permitted only for users RT and SYSTEM.
2. Access to files in RT-programs is as user RT.
3. RT-opened files cannot be set permanently open.
4. See rules 2-6 of @OPEN-FILE.

**Example:**

@RTOPEN-FILE F-1, RW

FILE NUMBER IS: 100

@

The file F-1:SYMB is opened for read and write access for all RT programs. The file number is 100<sub>8</sub>.

**@SAVE-DIRECTORY**

Same as @COPY-DEVICE.

**@SCHEDULE**

See also @RESERVE-FILE.

**Function:**

Reserve devices to be used by a job.

**Format:**

@SCHEDULE <logical device no.> ,....., <logical device no.>

**Parameters:**

<logical device no.>  
devices to be reserved (OCT).

**Rules:**

1. Numbers must be in ascending order.
2. Permitted for all users. Used especially in batch jobs, but may also be useful for time-sharing, eg. when further operations depend device availability.
3. If a specified device is reserved by another user, the calling user must wait until the device is released.
4. To prevent deadlock, all device numbers must occur in @SCHEDULE before they are used.
5. The devices may be released explicitly or they are automatically released at log out.

**Example:**

@SCHEDULE 2, 3, 4, 5

The batch job reserved logical device numbers 2 (paper tape reader), 3 (paper tape punch), 4 (card reader) and 5 (line printer).

## @ SCRATCH-OPEN

See also @OPEN-FILE, OPEN (MON 50) and @CONNECT-FILE.

### Function:

Open a file as a scratch file.

### Format:

@SCRATCH-OPEN <file name> , <access type>

### Parameters:

<file name>

(DEF type = SYMB).

<access type>

see @OPEN-FILE.

### Rules:

1. Permitted for all users.
2. The file is kept permanently opened. It is closed when logging out or by @CLOSE-FILE <file no.> but not @CLOSE-FILE 1. When closed, only the first 32K words are retained.
3. @CLOSE-FILE 2 also closes scratch files.
4. See rules 2-6 of @OPEN-FILE.

### Example:

@SCRATCH-OPEN SCRATCH:DATA, RW

FILE NUMBER IS: 101

The file SCRATCH:DATA is opened as a scratch file with read and write access.

## @SET

See also SET (MON 101). See also related @ABSET.

**Function:**

Enter an RT program in the time queue to be transferred to the execution queue after a specified period.

**Format:**

@SET <program> , <no. of time units> , <time unit>

**Parameters:**

<program>

RT-program name or address of RT-description (OCT, DEF = user's terminal background program).

<no. of time units>

the number of time units the program is to stay in the time queue.

<time unit>

1 = basic time units (normally 20 milliseconds)

2 = seconds

3 = minutes

4 = hours

**Rules:**

1. Permitted only for users RT and SYSTEM.
2. If the program is already in the time queue, it is removed and reinserted according to the new specifications.
3. If a program is put in the time queue by @SET, clock adjustment (@CLADJ) does not affect waiting time.

**Example:**

@SET KLOKK 5, 3

The RT program KLOKK is set up for execution in 5 minutes.

## @SET-AVAILABLE

See also related @SET-UNAVAILABLE.

**Function:**

Make the system available to terminal users.

**Format:**

@SET-AVAILABLE

**Parameters:**

None.

**Rules:**

1. Permitted only for user SYSTEM.
2. After starting SINTRAN, the system is unavailable until @SET-AVAILABLE is given.

## ① SET-BLOCK-POINTER

See also SETBL (MON 77). See also related @SET-BYTE-POINTER and @SET-BLOCK-SIZE.

### Function:

Set the byte pointer to the beginning of a block.

### Format:

@SET-BLOCK-POINTER <file no.>, <block no.>

### Parameters:

<file no.>  
logical device number of an opened mass-storage file (OCT:100-121).  
<block no.>  
(DEC, DEF = 0).

### Rules:

1. Permitted for all users.
2. The byte pointer is set to the product of <block no.> and block size (in bytes).

### Example:

@SET-BLOCK-SIZE 101, 512

@SET-BLOCK-POINTER 101, 1

The byte pointer is set to  $2000_8 = 1024_{10}$ .

## ① SET-BLOCK-SIZE

See also SETBS (MON 76). See also related @RFILE and @WFILE.

### Function:

Set the block size (in words) of an opened file.

### Format:

@SET-BLOCK-SIZE <file no.>, <block size>

### Parameters:

<file no.>  
file number of an opened mass-storage file (OCT:100-121).  
<block size>  
number of words per block (DEC:  $\geq 1$ ).

### Rules:

1. Permitted for all users.
2. Valid only as long as the file is opened; the next time the file is opened, the block size is by default  $256_{10}$  words.

### Example:

@SET-BLOCK-SIZE 101, 128

The block size of file with device number  $101_8$  is set to  $128_{10}$  words.

## @ SET-BYTE-POINTER

See also SETBL (MON 77). See also related @SET-BLOCK-POINTER.

### Function:

Set byte pointer to be used by the next INBT/OUTBT call on the specified mass-storage file.

### Format:

@SET-BYTE-POINTER <file no.> , <byte no.>

### Parameters:

<file no.>

file number of an opened mass-storage file (OCT:100-121).

<byte no.>

(OCT:0 — maximum byte pointer, DEF = 0).

### Rules:

Permitted for all users.

### Example:

@SET-BYTE-POINTER 101, 7

The byte pointer of file 101<sub>8</sub> is set to byte 7<sub>8</sub> relative to the start of the file (first byte is byte 0).

## @ SET-DEFAULT-DIRECTORY

See also related @ENTER-DIRECTORY.

### Function:

Define a directory as a default directory.

### Format:

@SET-DEFAULT-DIRECTORY <directory name>

### Parameters:

<directory name>

directory entered.

### Rules:

1. Permitted only for user SYSTEM when disk is specified, for any user when floppy disk.
2. Main directory is always default directory.
3. Several directories can be default directory, but a user should not have space in more than one default directory. If he has, he must always specify directory name when accessing a file.
4. The directory specified must be entered.

### Example:

@SET-DEFAULT-DIRECTORY PACK-TWO

PACK-TWO is defined as default directory.

## @ SET-DEFAULT-FILE-ACCESS

See also @SET-FILE-ACCESS. See also related @SET-FRIEND-ACCESS.

### Function:

Change the default file access for the current user. This is the public, friend and owner access to a file when it is created.

### Format:

@SET-DEFAULT-FILE-ACCESS <public access>,<friend access>,<owner access>

### Parameters:

<public access>

<friend access>

<owner access>

see @SET-FILE-ACCESS, rule 2 (DEF = no change).

### Rules:

1. Permitted for all users .
2. When the system is generated, default file access is public READ, friend READ, WRITE and APPEND, and all access for owner.

### Example:

@SET-DEFAULT-FILE-ACCESS N, RWA, RWACD

The default file access for public is changed to no access.

## @SET-ERROR-DEVICE

See also @GET-ERROR-DEVICE.

**Function:**

Change the destination of system error messages to the specified terminal.

**Format:**

@SET-ERROR-DEVICE <logical device no.>

**Parameters:**

<logical device no.>

device to which errors will be sent (DEC).

**Rules:**

1. Permitted only for user SYSTEM.
2. The messages are normally written on terminal 1.
3. The command is valid until next @SET-ERROR-DEVICE or until start from SAVE area. @RESTART-SYSTEM (MASTER CLEAR, LOAD) does not change the error device. Messages from the SINTRAN START program are written on the current error device.

**Example:**

@SET-ERROR-DEVICE 9

The messages are written on terminal 2. (Terminal no.2 has logical device no. 9, see Appendix C.)



## @ SET-FILE-ACCESS

See also @SET-DEFAULT-FILE-ACCESS. See also related @SET-FRIEND-ACCESS.

### Function:

Set public, friend and owner access of a specified file.

### Format:

@SET-FILE-ACCESS <file name>,<public access>,<friend access>,<own access>

### Parameters:

<file name>

(DEF type = SYMB).

<public access>

<friend access>

<own access>

a combination of access types, see below (DEF = no change).

### Rules:

1. Permitted for all users with directory access to the file.
2. The access type can be any relevant combination of the characters (except N which must stand alone):
  - R — read permitted
  - W — write permitted
  - A — append permitted (write from maximum byte pointer)
  - C — common access permitted (access by more than one user)
  - D — directory access permitted (the file may be created, deleted, legal access mode changed and new version created)
  - N — no access permitted
3. @FILE-STATISTICS can be used to check the access types for the file.

### Example:

@SET-FILE-ACCESS F-1:DATA N,N,RWACD

Public and friends have no access to the file F-1:DATA. Owner has total access.

## @SET-FRIEND-ACCESS

See also related @SET-DEFAULT-FILE-ACCESS, @SET-FILE-ACCESS and @CREATE-FRIEND.

### Function:

Specify access for a friend to the terminal user's files.

### Format:

@SET-FRIEND-ACCESS <friend name>, <access type>

### Parameters:

<friend name>

a name previously created as a friend.

<access type>

see @SET-FILE-ACCESS, rule 2 (DEF = no change).

### Rules:

1. Permitted for all users.
2. The permitted access to a file from a friend is the intersection of this friend's access and the general friend access of this specific file (see example below).

### Example:

(Assume friend access to file F-1:DATA to be WA.)

@SET-FRIEND-ACCESS BUDDY,W

Access to file F-1:DATA for user BUDDY is now W. That is, it may now only be opened with write access.

## @SET-MAIN-DIRECTORY

### Function:

Define a directory as main directory.

### Format:

@SET-MAIN-DIRECTORY

### Parameters:

<directory name>

### Rules:

1. Users in this directory need not be present in any other directory.
2. Up to 16 directories can be defined as main.
3. It is illegal to set directories on floppy disk as main.

There are some limitations, eg. two users belonging to two different main directories cannot be friends.

**@ SET-MEMORY-CONTENTS**

See also @LOOK-AT MEMORY.

**Function:**

Set area of user memory (terminal virtual memory) to a specific content.

**Format:**

@SET-MEMORY-CONTENTS <contents>,<low address>,<high address>

**Parameters:**

<contents>

(OCT, DEF = 0).

<low address>

<high address>

limits of memory area. Area includes limits (OCT, DEF = 0).

**Rules:**

Permitted for all users.

**Example:**

@SET-MEMORY-CONTENTS 124000,,177777

Set entire 64K to 124000<sub>8</sub>.

**@ SET-NUMBER-OF-PRINT-COPIES**

See also related @APPEND-SPOOLING-FILE.

**Function:**

Change the number of print copies desired for a file in the spooling queue.

**Format:**

@SET-NUMBER-OF-PRINT-COPIES <peripheral file name>,<file name>,<no. of copies>

**Parameters:**

<peripheral file name>

spooling device.

<file name>

a file in the spooling queue.

<no. of copies>

(DEC: ≥ 1).

**Rules:**

1. Permitted only for user SYSTEM and the user who appended the file.
2. If more than one entry in the queue matches <file name> only the first one is affected.

**Example:**

@SET-NUMBER-OF-PRINT-COPIES LINE-PRINTER, F-1, 8

Change the desired number of print copies to 8 of file F-1 in spooling queue to LINE-PRINTER.

## @ SET-PERIPHERAL-FILE

See also @SET-TERMINAL-FILE. See also related @SET-FILE-ACCESS.

**Function:**

Associate a file name with a logical device number.

**Format:**

@SET-PERIPHERAL-FILE <file name>,<logical device no.>

**Parameters:**

<file name>

if this is a new file the name must be given in quotes.

<logical device no.>

(OCT).

**Rules:**

1. Permitted only for user SYSTEM.
2. If a file is created in the command, default type is void (empty string).

**Example:**

@SET-PERIPHERAL-FILE 'LINE-PRINTER',5

The file LINE-PRINTER is created and associated with logical device number 5.

**@ SET-PERMANENT-OPENED**

See also related @OPEN-FILE, @CONNECT-FILE and @CLOSE-FILE.

**Function:**

Open a file so that it is not closed by @CLOSE -1 or the corresponding monitor call.

**Format:**

@SET-PERMANENT-OPENED <file no.>

**Parameters:**

<file no.>

file number from @OPEN-FILE or @CONNECT-FILE (OCT).

**Rules:**

1. Permitted for all users.
2. The file must be opened.
3. The file is closed by @CLOSE-FILE <file no.> or @CLOSE-FILE -2.

**Example:**

@OPEN-FILE OLE:DATA,RW

FILE NUMBER IS: 101

@SET-PERMANENT-OPENED 101

@CLOSE -1

The file OLE:DATA is still open.

## @ SET-SPOOLING-FORM

See also related @LIST-SPOOLING-FORM, @APPEND-SPOOLING-FILE and SPCLO (MON 40).

### Function:

Define an identification key to be compared to the user text in the spooling queue entries. The spooling program only prints the files with this user text or files with no text at all. The command is useful for sorting out files to be printed on the same form.

### Format:

@SET-SPOOLING-FORM <peripheral file name> , <spooling form identifier>

### Parameters:

<peripheral file name>

a spooling device.

<spooling form identifier>

any printable characters terminated by CR. (DEF = reset identification to a null string.)

### Rules:

Permitted only for user SYSTEM.

### Example:

@SET-SPOOLING-FORM LINE-PRINTER, FORM -1

Only spooling entries with the user message FORM -1 on LINE-PRINTER are printed.

## @ SET-TEMPORARY-FILE

See also related @FILE-STATISTICS.

### Function:

Define the contents of a file as temporary. The contents are deleted when the file is closed after it has been accessed for read. It is retained as an object entry.

### Format:

@SET-TEMPORARY-FILE <file name>

### Parameters:

<file name>  
an indexed file (DEF type = DATA).

### Rules:

1. Permitted for all users.
2. The file must not have been written to before the command is given.
3. Temporary files are useful as spooling files.
4. @FILE-STATISTICS show whether a file is temporary or not.

### Example:

@SET-TEMPORARY-FILE F-1;2

Version 2 of the file F-1 is defined as temporary file.

## @ SET-TERMINAL-FILE

See also @SET-PERIPHERAL-FILE.

### Function:

Set the name of terminal.

### Format:

@SET-TERMINAL-FILE <file name>

### Parameters:

<file name>  
to be used for terminal name.

### Rules:

1. Permitted only for user SYSTEM.
2. The name is set for all users.
3. Several files can be used as terminal files; any of the filenames can be used to write to the terminal.

### Example:

@SET-TERMINAL-FILE ' ' TERMINAL ' '

The new name is TERMINAL.

## @ SET-TERMINAL-TYPE

See also MSTTY (MON 17). See also related @GET-TERMINAL-TYPE.

**Function:**

Associate a terminal type number with a terminal (logical device number).

**Format:**

@SET-TERMINAL-TYPE <logical device no.>, <terminal type>

**Parameters:**

<logical device no.>

must be a terminal (DEC, DEF = user's terminal).

<terminal type>

(DEC: -32768 to +32767, DEF = 0).

**Rules:**

1. Permitted for all users but only user SYSTEM may specify terminals other than their own.
2. Norsk Data's standard for terminal types is shown in Appendix B.

**Example:**

@SET-TERMINAL-TYPE, -1

The user's terminal type is set to -1.

## @ SET-UNAVAILABLE

See also related @SET-AVAILABLE.

**Function:**

Set the system available only for terminal 1 (console terminal).

**Format:**

@SET-UNAVAILABLE <text>

**Parameters:**

<text>

a string of printable characters terminated by CR. When anyone tries to log in the message "SYSTEM UNAVAILABLE" and the <text> are output. The text should be terminated by \$ which is translated to CR, LF on output.

**Rules:**

Permitted only for user SYSTEM.

**Example:**

@SET-UNAVAILABLE DOWN FOR MAINT. \$ AVAILABLE 11:30\$

When anyone tries to log in on a terminal other than terminal 1 the following message is output:

```
SYSTEM UNAVAILABLE
DOWN FOR MAINT.
AVAILABLE 11:30
```



## @ SET-USER-PARAMETERS

### Function:

Set the five user parameters in SINTRAN.

### Format:

@ SET-USER-PARAMETERS

P1:

P2:

P3:

P4:

P5:

### Rules:

1. These parameters can be set and read by two monitor calls PASET (56) and PAGET (57).
2. Some of these parameters are set by SINTRAN if termination handling is enabled and a program is terminated with escape, MON 65 or a fatal error.

## @ SINTRAN-SERVICE-PROGRAM

### Function:

Activate service program for maintenance of resident memory, memory image and SINTRAN save area.

### Format:

@ SINTRAN-SERVICE-PROGRAM

See subcommands on the following pages.

### Parameters:

None.

### Rules:

1. Permitted only for user SYSTEM and for one terminal at a time.
2. If subcommands are used in a MODE or batch file each command must be preceded by @.
3. All parameters of the subcommands are default octal. A number terminated with a D (nnnD) specifies a decimal number.
4. Cannot be restarted with @CONTINUE.

## @ SINTRAN-SERVICE-PROGRAM

### \* ASCII-DUMP

See also \*OCTAL-DUMP.

#### Function:

Dump an area as ASCII characters.

#### Format:

\*ASCII-DUMP <area> [, <segment no.> ], <low address> , <high address> ,  
<output file>

#### Parameters:

See \*OCTAL-DUMP.

#### Rules:

Each line of the dump consists of a maximum of 64 characters. Control characters (LF, CR etc.) are output as printable characters. I.e. CR on TERMINAL generates a carriage return on the device.

## @ SINTRAN-SERVICE-PROGRAM

### \* CHANGE-BUFFER-SIZE

#### Function:

Change the length of the ring buffer of a logical device. Total buffer area is limited depending on system configuration; if a substantial increase in size is desired, calculations should be made.

#### Format:

\*CHANGE-BUFFER-SIZE <logical device no.> , <input or output> , <buffer size> , <image?> , <save-area?>

#### Parameters:

<logical device no.>  
(OCT).  
<input or output>  
<buffer size>  
length of ring buffer in words or bytes (OCT).  
<image?>  
<save-area?>  
YES = modify area.  
NO = do not modify (DEF = NO).

## @SINTRAN-SERVICE-PROGRAM

### \*CHANGE-DATAFIELD

**Function:**

Change value of displacements in datafields.

**Format:**

\*CHANGE-DATAFIELD <logical unit no.>,<input/output>,<memory?>,  
<image?>,<save-area?>  
Subcommands (see below)

**Parameters:**

<logical unit no.>  
(OCT).

<input or output>

input	=	input part.
output	=	output part.

<memory?>

<image?>

<save-area?>

YES	=	modify the area.
NO	=	do not modify area (DEF = NO).

**Rules:**

1. The command must not be used in batch jobs.
2. Subcommands for patching must follow rules 3, 4 and 6 of @LOOK-AT. The address specified must be a relative octal address within the datafield or the symbolic name of an item within the datafield.
3. The symbolic names available are:

AERRB	FLAGB	SERRB
ARG	FYLLE	SETDV
		SSREF
BHOLD	HDEV	STDEV
BLSZ	HENTE	STDRIV
BRKTAB	HSTAT	STRSEG
BSTATE		
BUFST	IBLOAD	TACNS
BWLINK	ICORAD	TACOUNT
	IFUNC	TERM
CARG	IMAXBHOLD	TMR
CFREE	IMAXW	TMSUB
CHARI	IOLOG	TRG
CNTREG	IOTRANS	TRLREG
COMFL	IRETW	TSPEED
CONVTAB	ISTATE	TSTATE
CTRG		TTMR
CTTYP	LAST	TYPRING
CXRG		
	MAX	WERRB
DBADR	MAXBHOLD	WFLAG
DBPROG	MFUNC	
DERROR	MINBHOLD	XRG
DFLAG	MLINK	
DFOPP	MTRANS	
DRG		
DRIVER	RESLINK	
DRT	RTRES	
ECHOTAB		
ERCNT		

**Example:**@SINTRAN-SERVICE-PROGRAM

\*CHANGE-DATAFIELD 36, I, Y, Y, Y

```

TSPEED/177777  177777  177777
              0      0      0

```

\*EX

®

The baud-rate (speed) for terminal 36 on a normal interface is changed to 9600 baud.

## @ SINTRAN-SERVICE-PROGRAM

### \*CHANGE-VARIABLE

#### Function:

Examine and change system variables, which may be single variables or arrays. For some variables, the memory image and/or save area may be affected. For some variables resident memory may be affected; this is indicated by an M in the rules below.

#### Format:

\*CHANGE-VARIABLE <variable name> [, <index> ], <value> [, <memory?> ]  
<image?> , <save-area?>

#### Parameters:

<variable name>

see rule 1.

<index>

only if <variable name> is an array (OCT). See rule 1.

<value>

new value (OCT, DEF = old value).

<memory?>

YES = modify area, NO = do not modify. Specified only if resident memory is to be changed.

<image?>

<save-area?>

YES = modify area.

NO = do not modify (DEF = NO).

#### Rules:

The following are legal values of <variable name>. The index range is specified for arrays.

BGLPAGE

Last logical page number legal for background programs.

BGFPAGE

First logical page number legal for background programs.

CACHLIM

Change the CACHE-INHIBIT-LIMIT register of NORD-10. The most significant byte is the upper limit and the least significant byte the lower limit.

CCFPAGE

First legal logical page for RTCOMMON.

CCLPAGE

Last legal logical page for RTCOMMON.

CNVRT (M,  $0 \leq \text{index} \leq 13$ )

Address of logical unit tables.

ENDCOR

Upper address of resident memory.

EXTDS (M,  $0 \leq \text{index} \leq 3$ )

Address of the extended ident code tables.

FIXMAX (M)

Maximum number of pages which can be fixed in memory simultaneously.

IDNTS (M,  $0 \leq \text{index} \leq 3$ )

Address of the ident code tables.

IMASK

Value is a mask to be used for enabling internal interrupts (TRR IIE).

## LCACHLIM

Change the lower limit of the CACHE-INHIBIT-LIMIT of NORD-100.

## LOADI (M)

= 0: RT loader shall initialize RTFIL.

≠ 0: RT loader will not initialize RTFIL.

## MAXP (M)

Maximum number of pages in memory for a demand segment.

## RTFPAGE

First legal logical page number for RT programs on page table 1.

## RTLPAGE

Last legal logical page number for RT programs on page table 1.

TABLES (M,  $0 \leq \text{index} \leq 3$ )

Address of timer, background, batch and RTCOMMON table (CCTAB).

TMCTAB (M,  $0 \leq \text{index} \leq 107$ )

Monitor call types (see \*DEFINE-USER-MONITOR-CALL). The TMCTAB array is a byte array ( $e_0, e_1, \dots, e_{217}$ , octal index). The parameter <index> is a word index. Thus two bytes are changed by one command. Eg. to change the type of byte element  $e_3$  to 1 when  $e_2$  is 16, use

\*CHANGE-VARIABLE TMCTAB, 1, 416, ...

## UCACHLIM

Change the upper limit of the CACHE-INHIBIT-LIMIT of NORD-100.

## UNAFIAG (M)

Flag set ≠ 0 if system is unavailable (@SET-UNAVAILABLE).

## USEGADR

Address of first free entry in segment table.

## UZEROFLAG (M)

= 0: the user's working area (background segment) will not be changed on each @LOGOUT.

≠ 0: the area is set to all zeros on each @LOGOUT. All pages released from a file by @DELETE-FILE are cleared.

## @ SINTRAN-SERVICE-PROGRAM

### \*DEFINE-BASIC-TIME-UNIT

See also related \*DEFINE-TIME-SLICE.

#### Function:

Set the length of the basic time unit in milliseconds.

#### Format:

\*DEFINE-BASIC-TIME-UNIT <milliseconds>,<image?>,<save-area?>

#### Parameters:

<milliseconds>

(OCT).

<image?>

<save-area?>

YES = modify area.

NO = do not modify (DEF = NO).

#### Rules:

The standard basic time unit is 20 ms.

## @ SINTRAN-SERVICE-PROGRAM

### \*DEFINE-BATCH-SUPERVISOR

#### Function:

Define an RT program to be started each time a batch job is terminated.

#### Format:

\*DEFINE-BATCH-SUPERVISOR <program name>,<memory?>,<image?>,<save-area?>

#### Parameters:

<program name>

RT description address or RT program name.

<memory?>

<image?>

<save-area?>

YES = modify area.

NO = do not modify (DEF = NO).

#### Rules:

None.

## @SINTRAN-SERVICE-PROGRAM

### \*DEFINE-RTCOMMON-SIZE

**Function:**

Define the size of the RTCOMMON area in pages.

**Format:**

\*DEFINE-RTCOMMON-SIZE <no. of pages> , <first physical page> ,  
<image?> , <save-area?>

**Parameters:**

<no. of pages>

new size (OCT).

<first physical page>

physical page where RT common should start. In addition to the pages generated for the system, only 8 pages can be added. (DEF = upper end of physical memory.)

<image?>

<save-area>

YES = modify area.

NO = do not modify (DEF = NO).

**Rules:**

Size can be increased by up to 10K in addition to the RT common size defined at system generation.



## @ SINTRAN-SERVICE-PROGRAM

### \* DEFINE-SEGMENT-FILE

#### Function:

Associate a segment file number with a segment file name for the RT-loader.

#### Format:

DEFINE-SEGMENT-FILE <memory?>, <save-area?>, <segment file no.> ,  
<segment file name> [, <redefine segment file?> ]

#### Parameters:

<memory?>

<save-area?>

YES = modify area.

NO = do not modify (DEF = NO).

<segment file no.>

(OCT: 0 - 3).

<segment file name>

any existing contiguous file.

<redefine segment file?>

only if <segment file no.> is already defined. Specify YES or NO  
(DEF = NO).

#### Rules:

1. Users SYSTEM and RT must have read and write access to <segment file name>.
2. The mass-storage address of the last page in segment file must be  $\leq 177777_8$ .
3. The command must not be used in batch jobs.
4. Redefine segment file only if <segment file no.> is already defined.

## @ SINTRAN-SERVICE-PROGRAM

### \*DEFINE-TIME-SLICE

See also related \*DEFINE-BASIC-TIME-UNIT.

#### Function:

Define the time slice of the background programs for terminal control and batch processors. The unit used in parameters is 10 basic time units.

#### Format:

```
*DEFINE-TIME-SLICE  <HLTIME>,<MLTIME>,<LTIML>,<LTIMU> ,
<image?> , <save-area?>
```

#### Parameters:

<HLTIME>

no. of basic units on high priority (OCT).

<MLTIME>

no. of basic units on medium priority (OCT).

<LTIML>

lower limit of basic units on low priority (OCT).

<LTIMU>

upper limit of basic units on low priority (OCT).

<image?>

<save-area?>

YES = modify area.

NO = do not modify (DEF = NO).

#### Rules:

The system is generated with standard values. They are:

<HTIME>	12
<MLTIME>	24
<LTIML>	24
<LTIMU>	100

## **@SINTRAN-SERVICE-PROGRAM**

### **\*DEFINE-TITLE**

#### **Function:**

Define string to be output in addition to system version string at log in on a terminal or as part of the spooling header.

#### **Format:**

\*DEFINE-TITLE <text>, <memory?>, <save-area?>

#### **Parameters:**

<text>

any printable characters terminated by an apostrophe ('). \$ is translated to CR, LF on output. CR on input is ignored. ( ' specifies no text.)

<memory?>

<save-area?>

YES = change area.

NO = do not change (DEF = NO).

#### **Rules:**

1. The maximum is 120 characters including apostrophe.
2. The text is printed by @LIST-TITLE and when "rub-out" (or DEL) is typed during remote operation.
3. The command must not be used in batch jobs.

#### **Example:**

\*DEFINE-TITLE \$NORD 10.54 \$ 810817 \$'

.

The text will be output as:

NORD 10.54

810817

## @ SINTRAN-SERVICE-PROGRAM

### \*DEFINE-USER-MONITOR-CALL

#### Function:

Define number, entry point address and type of user monitor call. The code should be assembled using FMAC or DMAC.

#### Format:

\*DEFINE-USER-MONITOR-CALL <monitor call no.> , <start address> ,  
<type> , <memory?> , <image?> , <save-area?>

#### Parameters:

<monitor call no.>  
(OCT).  
<start address>  
entry point of routine (OCT).  
<type>  
1 = callable only from RT programs.  
16 = callable from RT and background programs  
(OCT, DEF = 1).  
<memory?>  
<image?>  
<save-area?>  
YES = modify area.  
NO = do not modify (DEF = NO).

## @ SINTRAN-SERVICE-PROGRAM

### \*DEFINE-USER-RESTART-PROGRAM

See also \*DEFINE-USER-RESTART-SUBROUTINE.

#### Function:

Define the user restart RT program to be started by SINTRAN after each "power fail - restart".

#### Format:

\*DEFINE-USER-RESTART-PROGRAM <program> , <memory?> , <image?> ,  
<save-area?>

#### Parameters:

<program>  
RT description address (OCT), or RT program name.  
<memory?>  
<image?>  
<save-area?>  
YES = modify area.  
NO = do not modify (DEF = NO).

#### Rules:

None.

## @ SINTRAN-SERVICE-PROGRAM

### \*DEFINE-USER-RESTART-SUBROUTINE

See also \*DEFINE-USER-RESTART-PROGRAM.

#### Function:

Define the start address of a user restart subroutine to be called by SINTRAN after "power fail - restart".

#### Format:

```
*DEFINE-USER-RESTART-SUBROUTINE  <restart
address>,<memory?>,<image?>,<
save-area?>
```

#### Parameters:

```
<restart address>
    entry point of the subroutine (OCT).
<memory?>
<image?>
<save-area?>
    YES    =    modify area.
    NO     =    do not modify (DEF = NO).
```

#### Rules:

The subroutine must reside in resident memory.

## @ SINTRAN-SERVICE-PROGRAM

### \*DEFINE-USER-START-SUBROUTINE

#### Function:

Define the address of a user start subroutine executed when the system is started.

#### Format:

```
*DEFINE-USER-START-SUBROUTINE  <start address>,<image?>,<
save-area?>
```

#### Parameters:

```
<start address>
    entry point of subroutine (OCT).
<image?>
<save-area?>
    YES    =    modify area.
    NO     =    do not modify (DEF = NO).
```

#### Rules:

The subroutine must reside in resident memory or on the operator communication segment (segment 3).

## **@ SINTRAN-SERVICE-PROGRAM**

### **\*DELETE-SEGMENT-FILE**

**Function:**

Set a segment file unavailable for the RT loader. The file is not deleted from the directory. (See @DELETE-FILE.)

**Format:**

\*DELETE-SEGMENT-FILE <memory?>, <image?>, <save-area?>, <segment file no.>

**Parameters:**

<memory?>

<image?>

<save-area?>

YES = modify area.

NO = do not modify (DEF = NO).

<segment file no.>

(OCT: 0-3).

**Rules:**

A defined segment file must not be deleted (@DELETE-FILE) without being undefined with \*DELETE-SEGMENT-FILE.

## **@ SINTRAN-SERVICE-PROGRAM \*DUMP-RT-DESCRIPTION**

See also @LIST-RT-DESCRIPTION.

### **Function:**

Dump the symbolic names and contents of locations in an RT description.

### **Format:**

\*DUMP-RT-DESCRIPTION <program>,<area>,<output file>

### **Parameters:**

<program>

RT description address (OCT) or RT program name.

<area>

memory, image or save-area.

<output file>

destination of the dump (DEF = TERMINAL).

### **Rules:**

None.

## **@ SINTRAN-SERVICE-PROGRAM \*DUMP-SEGMENT-TABLE-ENTRY**

See also @LIST-SEGMENT.

### **Function:**

Dump the symbolic name and contents of locations in a segment table entry.

### **Format:**

\*DUMP-SEGMENT-TABLE-ENTRY <segment no.>,<area>,<output file>

### **Parameters:**

<segment no.>

the segment to be dumped (OCT).

<area>

choose memory, image or save-area.

<output file>

destination of the dump (DEF = TERMINAL).

### **Rules:**

None.

### **Example:**

```
*DUMP-SEGMENT-TABLE-ENTRY 30,M,,
SEGLINK:                0
BPAGELINK:              0
LOGADR:                 1074
MADR:                   325
FLAG:                   162003
```

**@SINTRAN-SERVICE-PROGRAM**  
**\*EXIT****Function:**

Leave the service program and return to the SINTRAN command processor.

**Format:**

\*EXIT.

**Parameters:**

None.

**Rules:**

None.

**@SINTRAN-SERVICE-PROGRAM**  
**\*HELP**  
**\*LIST-SERVICE-COMMANDS****Function:**

List all subcommands available in @SINTRAN-SERVICE-PROGRAM.

**Format:**

\*HELP <output file>

\*LIST-SERVICE-COMMANDS <output file>

**Parameters:**

<output file>

destination of the list (DEF = TERMINAL).

**Rules:**

The two commands are identical.



## @ SINTRAN-SERVICE-PROGRAM

### \*INITIALIZE-SYSTEM-SEGMENT

#### Function:

Fetch a new copy of the system segment for a terminal. The segment is transferred from the save area to the segment file.

#### Format:

\*INITIALIZE-SYSTEM-SEGMENT <logical device no.>

#### Parameters:

<logical device no.>

the terminal's logical device number (OCT).

#### Rules:

This command will only be executed if the terminal's background processor is passive; otherwise the message "background program active" is given.

## @ SINTRAN-SERVICE-PROGRAM

### \*INSERT-IN-BACKGROUND-TABLE

### \*REMOVE-FROM-BACKGROUND-TABLE

#### Function:

Insert (remove) a device into (from) the background table.

#### Format:

\*INSERT-IN-BACKGROUND-TABLE <logical device no.>, <input/output>, <memory?>, <image?>, <save-area?>

\*REMOVE-FROM-BACKGROUND-TABLE <logical device no.>, <input/output>, <memory?>, <image?>, <save-area?>

#### Parameters:

<logical device no.>

(OCT).

<input/output>

choose input or output.

<memory?>

<image?>

<save-area?>

YES = modify area.

NO = do not modify (DEF = NO).

#### Rules:

For insert, there must be a free entry in the table.

**@ SINTRAN-SERVICE-PROGRAM**  
**\*INSERT-IN-EXTENDED-IDENT-TABLE**  
**\*REMOVE-FROM-EXTENDED-IDENT-TABLE**

**Function:**

Insert (remove) an entry in (from) the extended ident code table of a hardware interrupt level.

**Format:**

\*INSERT-IN-EXTENDED-IDENT-TABLE <level>,<logical device no.>,  
<input/output>,<ident code>,<memory?>,<image?>,<save-area?>

\*REMOVE-FROM-EXTENDED-IDENT-TABLE <level>,<logical device no.>,  
<input/output>,<ident code>,<memory?>,<image?>,<save-area>

**Parameters:**

See \*INSERT-IN-IDENT-TABLE.

**Rules:**

None.

**@ SINTRAN-SERVICE-PROGRAM**  
**\*INSERT-IN-IDENT-TABLE**  
**\*REMOVE-FROM-IDENT-TABLE**

**Function:**

Insert (remove) an entry in (from) the ident code table of a hardware interrupt level.

**Format:**

\*INSERT-IN-IDENT-TABLE <level>, <logical device no.>, <input/output>,  
 <ident code>, <memory?>, <image?>, <save-area?>

\*REMOVE-FROM-IDENT-TABLE <level>, <logical device no.>,  
 <input/output>, <memory?>, <image?>, <save-area?>

**Parameters:**

<level>

Interrupt level (OCT).

<logical device no.>

(OCT: 1-77, 200—).

<input/output>

<ident>

new ident code (OCT).

<memory?>

<image?>

<save-area?>

YES = modify area.

NO = do not modify (DEF = NO).

**Rules:**

None.

**Example:**

\*INSERT-IN-IDENT-TABLE 12, 5, OUTPUT, 3

\*REMOVE-FROM-IDENT-TABLE 12, 5, OUTPUT

**@ SINTRAN-SERVICE-PROGRAM**  
**\*INSERT-IN-IOX-TABLE**  
**\*REMOVE-FROM-IOX-TABLE**

See also related @EXECUTE-IOX and EXIOX (MON 31).

**Function:**

Make a hardware device number available (unavailable) for @EXECUTE-IOX and EXIOX.

**Format:**

\*INSERT-IN-IOX-TABLE <hardware device no.> , <memory?> , <image?> ,  
 <save-area?>

\*REMOVE-FROM-IOX-TABLE <hardware device no.> , <memory?> ,  
 <image?> , <save-area?>

**Parameters:**

<hardware device no.>  
 (OCT).

<memory?>

<image?>

<save-area?>

YES = modify area.

NO = do not modify (DEF = NO).

**Rules:**

None.

**@ SINTRAN-SERVICE-PROGRAM**  
**\*INSERT-IN-LOGICAL-UNIT-TABLE**  
**\*REMOVE-FROM-LOGICAL-UNIT-TABLE**

**Function:**

Insert (remove) a logical unit in (from) the logical unit table.

**Format:**

\*INSERT-IN-LOGICAL-UNIT-TABLE <logical device no.>,<input/output>,  
 <datafield>

\*REMOVE-FROM-LOGICAL-UNIT-TABLE <logical device no.>,  
 <input/output>,<datafield>

**Parameters:**

<logical device no.>

(OCT: 1-77, 200—).

<input/output>

<datafield>

address of the datafield of device (OCT).

<memory?>

<save-area?>

YES = modify area.

NO = do not modify (DEF = NO).

**Rules:**

For insert there must be a free entry in the table for <logical unit no.>.

**Example:**

\*INSERT-IN-LOGICAL-UNIT-TABLE 5, INPUT, 14341

\*REMOVE-FROM-LOGICAL-UNIT-TABLE 6, OUTPUT

**@SINTRAN-SERVICE-PROGRAM**  
**\*INSERT-IN-TIME-SLICE**  
**\*REMOVE-FROM-TIME-SLICE**

See @PRIOR. **Function:**

Insert causes the background program priority period to be changed; remove causes the background program to be run at a fixed priority.

**Format:**

\*INSERT-IN-TIME-SLICE <logical device no.> , <memory?> , <image?> ,  
 <save-area?>

REMOVE-FROM-TIME-SLICE <logical device no.> , <memory?> , <image?> ,  
 <save-area?>

**Parameters:**

<logical device no.>  
     a terminal or batch processor (OCT).  
 <memory?>  
 <image?>  
 <save-area?>  
     YES     =     modify area.  
     NO      =     do not modify (DEF = NO).

**Rules:**

A background program not in the time slice system will run on a fixed priority; this is defined by @PRIOR.

**@ SINTRAN-SERVICE-PROGRAM**  
**\*INSERT-IN-TIMER-TABLE**  
**\*REMOVE-FROM-TIMER-TABLE**

**Function:**

Insert (remove) a logical device number in (from) the timer table.

**Format:**

\*INSERT-IN-TIMER-TABLE <logical device no.> , <input/output>

\*REMOVE-FROM-TIMER-TABLE <logical device no.> , <input/output>

**Parameters:**

<logical device no.>

(OCT: 1-77, 200—).

<input/output>

<memory?>

<save-area?>

YES = modify area.

NO = do not modify (DEF = NO).

**Rules:**

For insert there must be a free entry in the table.

**Example:**

\*INSERT-IN-TIMER-TABLE 5, OUTPUT

\*REMOVE-FROM-TIMER-TABLE 6, INPUT

**@ SINTRAN-SERVICE-PROGRAM**  
**\*INSERT-SPOOLING-HEADER**  
**\*REMOVE-SPOOLING-HEADER**

**Function:**

Cause the spooling program to insert (remove) the spooling header and trailer between each file output by the spooling system.

**Format:**

\*INSERT-SPOOLING-HEADER <spooling index> , <memory?> , <image?> ,  
 <save-area?>

\*REMOVE-SPOOLING-HEADER <spooling index> , <memory?> , <image?> ,  
 <save-area?>

**Parameters:**

<spooling index>

index of the spooling device in the system.

**Rules:**

None.

## @SINTRAN-SERVICE-PROGRAM

### \*LIST-ADDRESSES

**Function:**

List addresses of some global variables and entry points within the fixed part of SINTRAN.

**Format:**

\*LIST-ADDRESSES <output file>

**Parameters:**

<output file>

destination of the list (DEF = TERMINAL).

**Rules:**

None.

**Example:**

\*LIST-ADDRESSES.

RET:	1333
GET0:	1231
GET1:	1223
GET2:	1215
GET3:	1207
GET4:	1201
GET5:	1173
CCTAB:	45062
*	

## @SINTRAN-SERVICE-PROGRAM

### \*LIST-SERVICE-COMMANDS

See \*HELP.



## @ SINTRAN-SERVICE-PROGRAM

### \*OCTAL-DUMP

See also \*ASCII-DUMP.

#### Function:

Dump an area as octal numbers.

#### Format:

\*OCTAL-DUMP <area> [, <segment no.> ], <low address> , <high address> ,  
<output file>

#### Parameters:

<area>

choose memory, image, save-area or segment.

<segment no.>

only if <area> = segment (OCT).

<low address>

<high address>

address range to be dumped including limits (OCT, DEF = 0).

<output file>

destination of the dump (DEF = TERMINAL).

#### Rules:

Each output line dumps 10<sub>8</sub> words. The address of the first word on each line is given in a separate column.

#### Example:

\*OCTAL-DUMP MEMORY, , 1000, 1777, ,

Dump memory area 1000 to 1777 as octal numbers.

## @ SINTRAN-SERVICE-PROGRAM

### \*READ-BINARY

#### Function:

Read binary information from a file to a system area.

#### Format:

\*READ-BINARY <area> , <file name>

#### Parameters:

<area>

choose memory, image or save-area.

<file name>

(DEF type = :BPUN).

#### Rules:

None.

**@ SINTRAN-SERVICE-PROGRAM**  
**\*REMOVE-FROM-BACKGROUND-TABLE**

See \*INSERT-IN-BACKGROUND-TABLE.

**@ SINTRAN-SERVICE-PROGRAM**  
**\*REMOVE-FROM-EXTENDED-IDENT-TABLE**

See \*INSERT-IN-EXTENDED-IDENT-TABLE.

**@ SINTRAN-SERVICE-PROGRAM**  
**\*REMOVE-FROM-IDENT-TABLE**

See \*INSERT-IN-IDENT-TABLE.

**@ SINTRAN-SERVICE-PROGRAM**  
**\*REMOVE-FROM-IOX-TABLE**

See \*INSERT-IN-IOX-TABLE.

**@ SINTRAN-SERVICE-PROGRAM**  
**\*REMOVE-FROM-LOGICAL-UNIT-TABLE**

See \*INSERT-IN-LOGICAL-UNIT-TABLE.

**@ SINTRAN-SERVICE-PROGRAM**  
**\*REMOVE-FROM-TIMER-TABLE**

See \*INSERT-IN-TIMER-TABLE.

## @ SINTRAN-SERVICE-PROGRAM \*REMOVE-SINTRAN-COMMAND

### Function:

Remove a SINTRAN command (not file system, RT-loader or other subsystems).

### Format:

\*REMOVE-SINTRAN-COMMAND

### Parameters:

<command>  
<memory?>  
<save-area?>  
\*EX

)HENT is necessary to recover the command if it is removed in memory. If it is removed on save also the system must be reloaded to recover it.

## @ SINTRAN-SERVICE-PROGRAM \*REMOVE-SPOOLING-HEADER

See \*INSERT-SPOOLING-HEADER.

## @ SINTRAN-SERVICE-PROGRAM \*SET-CHANNEL-PRIORITY

### Function:

Set priority of a communication channel.

### Format:

\*SET-CHANNEL-PRIORITY <channel name>, <priority>, <memory?>,  
<image?>, <save-area?>

### Parameters:

<channel name>  
    peripheral file name of a channel.  
<priority>  
    new priority (OCT: 0-377).  
<memory?>  
<image?>  
<save-area?>  
    YES     =     modify area.  
    NO      =     do not modify (DEF = NO).

### Rules:

None.

## @SINTRAN-SERVICE-PROGRAM

### \*SET-CLOSED-SCRATCH-FILE-SIZE

#### Function:

Set the maximum number of pages to remain in a scratch file when it is closed.

#### Format:

\*SET-CLOSED-SCRATCH-FILE-SIZE <logical device no.>,<no. of pages>,  
<memory?>,<save-area?>

#### Parameters:

<logical device no.>  
the terminal associated with the scratch file (OCT).  
<no. of pages>  
maximum number of pages (multiple of 10<sup>6</sup>) (OCT).  
<memory?>  
<save-area?>  
YES = modify area.  
NO = do not modify (DEF = NO).

#### Rules:

The <size> for all scratch files is changed when the save-area is modified.

## @SINTRAN-SERVICE-PROGRAM

### \*SET-COMMAND-PROTECTION

#### Function:

Change the group of users who are allowed to use a SINTRAN III command.

#### Format:

\*SET-COMMAND-PROTECTION <command>,<protection>,<memory?>,  
<save-area?>

#### Parameters:

<command>  
any command except file system commands (can be abbreviated).  
<protection>  
PUBLIC = permitted for all users.  
RT = permitted for users RT and SYSTEM.  
SYSTEM = permitted only for user SYSTEM.  
<memory?>  
<save-area?>  
YES = modify area.  
NO = do not modify (DEF = NO).

#### Rules:

The command parameter can be abbreviated.

**@ SINTRAN-SERVICE-PROGRAM**  
**\*SET-SPOOLING-DEVICE-NUMBER**

**Function:**

Associate a spooling index with a logical device number.

**Format:**

\*SET-SPOOLING-DEVICE-NUMBER <spooling index>,<logical device no.>,<memory?>,<image?>,<save-area?>

**Parameters:**

<spooling index>

index of the spooling device in the system. The range is determined at system generation time (OCT).

<logical device no.>

logical device to be used for spooling (OCT).

<memory?>

<image?>

<save-area?>

YES = modify area.

NO = do not modify (DEF = NO).

**Rules:**

None.

**@ SINTRAN-SERVICE-PROGRAM**  
**\*START-XMSG**  
**\*STOP-XMSG**

See XMSG program description sheet and SINTRAN Communication Guide, ND-60.134.01 for details.

## @SINTRAN-SERVICE-PROGRAM

### \*SWAP-DIRECTORY-ENTRIES

#### Function:

Exchange two directories in the directory table.

#### Format:

\*SWAP-DIRECTORY-ENTRIES <directory index 1>,<directory index 2>,  
<save-area?>

#### Parameters:

<directory index 1>

<directory index 2>

index refers to the relative location in the directory table  
(OCT:  $\geq 0$ ).

<save-area?>

YES	=	modify area.
NO	=	do not modify.

## @SPOOLING-PAGES-LEFT

See also related @GIVE-SPOOLING-PAGES and @TAKE-SPOOLING-PAGES.

#### Function:

List the remaining number of pages that can be used by the spooling files.

#### Format:

@SPOOLING-PAGES-LEFT

#### Parameters:

None.

#### Rules:

1. Permitted for all users.
2. Should be less than the free space under user SYSTEM.

#### Example:

```
@SPOOLING-PAGES-LEFT
500 SPOOLING PAGES LEFT
@
```

## @ START-ACCOUNTING

See also related @INIT-ACCOUNTING, @STOP-ACCOUNTING, @START-RT-ACCOUNT and @STOP-RT-ACCOUNT.

### Function:

Start the accounting system but do not initiate the accounting file; if the file does not exist the command executes as for @INIT-ACCOUNTING.

### Format:

@START-ACCOUNTING <background> [, <RT> , <clear logged information> , <logging interval> ][, <ND-500> ]

### Parameters:

<background>

are background programs to be accounted.

<RT>

are user RT programs to be accounted.

<clear logged information>

whether information already logged in the RT accounting table should be cleared.

<logging interval>

number of seconds between dumps of RT accounting table on the file ACCOUNTS:DATA.

### Rules:

Permitted only for user SYSTEM.

## @ START-COMMUNICATION

See also related @STOP-COMMUNICATION.

### Function:

Initiate communication on a communication line. A configuration table showing corresponding channel and logical device numbers is sent to the remote computer. If this computer answers with its own configuration table within 12 seconds, the message COMMUNICATION ESTABLISHED is written. If no answer is received, the message NO REPLY is written and control is returned to the command processor. However, the system continues to retransmit the configuration table until @STOP-COMMUNICATION is given.

### Format:

@START-COMMUNICATION <line no.>

### Parameters:

<line no.>

(DEC, DEF = 1 if the system is configured with only one line).

### Rules:

1. Permitted only for users RT and SYSTEM.
2. To establish communication it must be started on both sides of the communication line.
3. The first communication line started causes NORDNET programs to be fixed in memory.

## @START-HISTOGRAM

See also related @DEFINE-HISTOGRAM, @DEFINE-SYSTEM-HISTOGRAM, @STOP-HISTOGRAM and @PRINT-HISTOGRAM.

**Function:**

Turn on sampling for the histogram.

**Format:**

@START-HISTOGRAM.

**Parameters:**

None.

**Rules:**

1. Permitted for all users.
2. The histogram must be defined (@DEFINE-HISTOGRAM).
3. The command is normally given before starting the program to be sampled.

## @START-PRINT

See also related @STOP-PRINT, @APPEND-SPOOLING-FILE and SPCLO (MON 40).

**Function:**

Resume printout of the current spooling file.

**Format:**

@START-PRINT <peripheral file name>

**Parameters:**

<peripheral file name>  
the spooling device.

**Rules:**

1. Permitted only for user SYSTEM and the user who appended the file to be printed.
2. Print can be stopped by:
  - a) @STOP-PRINT.
  - b) an automatic stop print defined by @DEFINE-SPOOLING-CONDITIONS.
  - c) closing a file with SPCLO (MON 40) specifying stop print.
  - d) specifying stop print in @APPEND-SPOOLING.



## @ START-PROGRAM-LOG

See also related @STOP-PROGRAM-LOG and @RT-PROGRAM-LOG.

**Function:**

Start the logging of RT and background programs. When used with @DEFINE-SYSTEM-HISTOGRAM produces a sampling distribution on various interrupt levels.

**Format:**

@START-PROGRAM-LOG <interrupts/sample>

**Parameters:**

<interrupts/sample>  
number of terminal interrupts per sample.

**Rules:**

Permitted only for user SYSTEM.

**Example:**

(on a 9600 baud rate terminal)

@START-PROGRAM-LOG 100

Samples are taken at a rate of approximately 10 samples/second.

## @ START-RT-ACCOUNT

**Function:**

Start accounting for RT programs.

**Format:**

@START-RT-ACCOUNT

**Parameters:**

<RT program>

**Rules:**

1. Each user RT program is associated with a project password.
2. Available for users SYSTEM and RT.
3. System RT programs cannot be logged.

## @START-SPOOLING

See also related @STOP-SPOOLING.

### Function:

Start the spooling program for a peripheral, which is reserved by the spooling program and prints the files already in the spooling queue and those put in later.

### Format:

@START-SPOOLING <peripheral file name>

### Parameters:

<peripheral file name>  
the spooling device.

### Rules:

1. Permitted only for user SYSTEM.
2. The number of pages given to spooling files is compared to the number of unused pages belonging to user SYSTEM. If the latter number is smaller, the number of pages given to spooling will be reduced accordingly (see also @SPOOLING-PAGES-LEFT).
3. If more than one version of the file is a peripheral file, the spooling programs for all peripheral versions of the file are started. One specific peripheral file can be selected by including a version number in the file name.
4. An error message appears if the <peripheral file name> is not the name of a peripheral or if no spooling program exists for the peripheral.
5. @RTENTER must be given before this command is executed.
6. If there are files in the spooling queue, @START-SPOOLING causes immediate output at the spooling device.
7. The first spooling process started causes the spooling program to be fixed in memory.

## @ STATUS

See also @LOOK-AT MEMORY.

**Function:**

Print the register contents of the background program.

**Format:**

@ STATUS

**Parameters:**

None.

**Rules:**

1. Permitted for all users.
2. The printout is as follows:
 

P = xxxxxx	program counter
X = xxxxxx	post-index register
T = xxxxxx	temporary register
A = xxxxxx	accumulator
D = xxxxxx	double accumulator
L = xxxxxx	subroutine link address register
S = xxxxxx	status register
B = xxxxxx	pre-index (base) register

## @ STOP-ACCOUNTING

See also related @INIT-ACCOUNTING and @START-ACCOUNTING.

**Function:**

Stop the accounting of system resources.

**Format:**

@STOP-ACCOUNTING <background> [, <RT> ][, <ND-500> ]

**Parameters:**

<background> (if started)  
 <RT> (if started)  
 <ND-500>

**Rules:**

1. Permitted only for user SYSTEM.
2. The accounting file is not affected.

## @STOP-COMMUNICATION

See also related @START-COMMUNICATION.

**Function:**

Turn off communication on a line.

**Format:**

@STOP-COMMUNICATION <line no.>

**Parameters:**

<line no.>

(DEC, DEF = 1 if the system is configured with only one line).

**Rules:**

1. Permitted only for users RT and SYSTEM.
2. Communication should be turned off on both sides of the communication line.
3. The NORDNET program (segment) is unfixed (UNFIX) when the last line is stopped.

## @STOP-HISTOGRAM

See also related @DEFINE-HISTOGRAM, @PRINT-HISTOGRAM and @START-HISTOGRAM.

**Function:**

Turn off sampling for the histogram.

**Format:**

@STOP-HISTOGRAM

**Parameters:**

None.

**Rules:**

1. Permitted for all users.
2. @STOP-HISTOGRAM is performed as part of @PRINT-HISTOGRAM.

## @ STOP-PRINT

See also @ABORT-PRINT. See also related @START-PRINT, @BACKSPACE-PRINT, @FORWARD-SPACE-PRINT and @RESTART-PRINT.

**Function:**

Stop the current printout and await further commands.

**Format:**

@STOP-PRINT <peripheral file name>

**Parameters:**

<peripheral file name>  
the spooling device.

**Rules:**

1. Permitted only for user SYSTEM and the user who appended the file being printed.
2. The current print buffer is finished before the printing stops (@ABORT-PRINT causes immediate stop).

## @STOP-PROGRAM-LOG

See also related @START-PROGRAM-LOG.

### Function:

Stop logging programs and print report on a file. The program names are printed and the percentage of the measured time during which the program has been active. If combined with @DEFINE-SYSTEM-HISTOGRAM, produces a list of sampling distribution on various interrupt levels.

### Format:

@STOP-PROGRAM-LOG <output file>

### Parameters:

<output file>

destination of the report (DEF = TERMINAL).

### Rules:

Permitted only for user SYSTEM.

### Example:

@STOP-PROGRAM-LOG,

	PERCENT	SAMPLES
DUMMY:	63	10258
STSIN:	00	0
RTERR:	00	0
RTSLI:	01	114
RWRT1:	00	0
RWRT2:	00	0
RWRT3:	00	0
SCOM1:	00	0
RCOM1:	00	2
SCOM2:	00	0
RCOM2:	00	0
BAK01:	00	0
BAK02:	00	0
BAK03:	14	2341
BAK04:	00	0
BAK05:	01	132
BAK06:	02	253
BAK07:	02	258
BAK08:	18	2949
BAK09:	00	7
BAK10:	00	0
BAK11:	00	0
...		

## @ STOP-RT-ACCOUNT

See related @START-RT-ACCOUNT, @INIT-ACCOUNTING and @START-ACCOUNTING. See also SINTRAN III System Supervisor manual, ND-30.003.

### Function:

Stop accounting for individual RT programs.

### Format:

@STOP-RT-ACCOUNT <RT program name>

### Parameters:

<RT program>

### Rules:

1. Each RT program is associated with a project password.
2. Available for users SYSTEM and RT.

## @ STOP-SPOOLING

See also related @START-SPOOLING.

### Function:

Stop the spooling program for a peripheral and release the peripheral from the spooling program.

### Format:

@STOP-SPOOLING <peripheral file name>

### Parameters:

<peripheral file name>  
the spooling device.

### Rules:

1. Permitted only for user SYSTEM.
2. Valid only if the queue is empty or the peripheral is stopped between two files (use @ABORT-PRINT to stop immediately).
3. Files can still be appended to the queue. The spooling program resumes printing the files in the queue when @START-SPOOLING is given.
4. The spooling program segment is unfixed (UNFIX) when the last spooling process is finished.

## @ STOP-SYSTEM

**Function:**

Stop the system.

**Format:**

@STOP-SYSTEM (Do not abbreviate.)

**Parameters:**

None.

**Rules:**

1. Permitted only for user SYSTEM.
2. All hardware registers are saved before the system goes into stop mode.
3. The system can be restarted by pressing the RESTART button on a NORD-10 (20! on ND-100) on the operator's panel. On restart logged in users can continue their programs. No start up procedure is necessary and no information is lost.
4. To turn off the system for a longer period:
  - a) Log out all users (preferably).
  - b) Press STOP and MASTER CLEAR on operator's panel.
  - c) Stop the disk only if temperature, energy, and/or noise conditions require it.The system should normally be left running, even over night. It should only be turned off for hardware maintenance, etc.



## @STOP-TERMINAL

See also @LOGOUT. See also related @SET-UNAVAILABLE.

**Function:**

Force @LOGOUT on a specific terminal.

**Format:**

@STOP-TERMINAL <logical device no.>

**Parameters:**

<logical device no.>  
must be a terminal (DEC).

**Rules:**

1. Permitted only for user SYSTEM.
2. The message **\*\*ABORTED BY SYSTEM\*\*** is printed on the specified terminal.
3. In some cases the background program hangs in an internal "escape off" state. @STOP-TERMINAL cannot then be used. Instead the other users should log off and SINTRAN should be restarted.
4. If the command has no effect on a terminal, @ABORT BAK can be used, but with care as in a few cases this may stop the system.

**Example:**

@STOP-TERMINAL 8  
Terminal 8 will be logged out.

## @TAKE-SPOOLING-PAGES

See also related @GIVE-SPOOLING-PAGES.

**Function:**

Decrease the total number of pages that can be used by the spooling files.

**Format:**

@TAKE-SPOOLING-PAGES <no. of pages>

**Parameters:**

<no. of pages>  
the number of pages to be removed (DEC).

**Rules:**

1. Permitted only for user SYSTEM.
2. The number of pages to be removed cannot be greater than the number of unused pages available for spooling files.

**Example:**

@TAKE-SPOOLING-PAGES 95  
95 unused spooling pages are removed.

## @TAKE-USER-SPACE

See also related @GIVE-USER-SPACE.

**Function:**

Decrease the total amount of space available to a user for files.

**Format:**

@TAKE-USER-SPACE [ <directory name> : ] <user name> , <no. of pages>

**Parameters:**

[ <directory name> : ] <user name>  
directory and user from whom space is taken (DEF directory is main directory for that user).  
<no. of pages>  
number of pages to be removed (DEC).

**Rules:**

1. Permitted only for user SYSTEM when disk is specified, for all users when floppy disk.
2. The number of pages to be removed cannot exceed the number of unused pages owned by the user.

**Example:**

@TAKE-USER-SPACE USER-ONE, 95  
95 unused pages are taken from the user USER-ONE in main his directory.

**@TERMINAL-MODE**

See also TERMO (MON 52).

**Function:**

Set the communication mode of the terminal.

**Format:**

@TERMINAL-MODE <CAPITAL LETTERS?>,<DELAY AFTER CR?>,<STOP ON FULL PAGE?>,<LOG OUT ON MISSING CARRIER?>

**Parameters:**

(DEF on all parameters = No change.)

## &lt;CAPITAL LETTERS?&gt;

- YES = all lower case letters are converted to upper case on input.
- NO = no conversion.

## &lt;DELAY AFTER CR?&gt;

- YES = dummy characters are printed after CR. This is necessary for some high speed hard copy terminals.
- NO = no dummy characters.

## &lt;STOP ON FULL PAGE?&gt;

- YES = the output stops after 20 lines if there has been no input during that period. A "bell" character is then output. Any remaining output continues as soon as a character (preferably NULL, SHIFT/CONTROL/P) is typed. This is useful for fast displays.
- NO = no stop.

## &lt;LOG OUT ON MISSING CARRIER?&gt;

- YES = if a "missing carrier" occurs on the connection between the terminal and the computer, the background program logs out.
- NO = the process on the terminal is stopped and the terminal is returned to command mode (indicated by @).

**Rules:**

1. Permitted for all users.
2. When logging in, all parameters are set to NO.

**Example:**

@TERMINAL-MODE,,,YES,,

Only the parameter <STOP ON FULL PAGE?> is changed to YES.

**@TERMINAL-STATUS**

See also related @WHO-IS-ON.

**Function:**

List the status of one or all active terminals. The information listed is:

LOG NO.: logical device number  
 MODE: COMMAND or USER (executing user program)  
 CPU MIN: CPU time used in minutes  
 OUT OF: total time logged on  
 LAST COMMAND: last SINTRAN command or last input line entered on terminal (only first 11 characters)

**Format:**

@TERMINAL-STATUS <logical device no.>,<interval>

**Parameters:**

<logical device no.>

must be a terminal (DEC, DEF = all active terminals).

<interval>

time interval in seconds between each time the listing is output, 0 = only one listing (DEC, DEF = 0).

**Rules:**

1. Permitted for all users.
2. Periodic output is terminated by pressing "escape".

**Example:**

@TERMINAL-STATUS 35, 5

(form feed is output)

LOG. NO	USER	MODE	CPU-MIN	OUT OF	LAST COMMAND
---------	------	------	---------	--------	--------------

35	USER ONE	USER	13	91	TERMINAL-ST
----	----------	------	----	----	-------------

The same output will be printed again after 5 seconds.

## @TEST-DIRECTORY

See also @REGENERATE-DIRECTORY.

**Function:**

Test directory for conflicting references to pages, incorrect user/object entries, incorrect bit file etc. If any are found, write error message. The bit file is rebuilt.

**Format:**

@TEST-DIRECTORY <directory name>

**Parameters:**

<directory name>  
an entered directory.

**Rules:**

1. Permitted only for user SYSTEM.
2. See rule 2 of @REGENERATE-DIRECTORY
3. The command works like @REGENERATE-DIRECTORY except that only the bit file is modified.
4. The command should run to completion; if not the bit file may be destroyed. Never press escape.
5. The FILE-SYSTEM-INVESIGATOR can be used instead of this command.

## @TIME-USED

See also @LOGOUT, @TERMINAL-STATUS and TUSED (MON 114).

**Function:**

Print CPU time and connect time used since log in or start of batch job.

**Format:**

@TIME-USED

**Parameters:**

None.

**Rules:**

Permitted for all users.

**Example:**

@TIME-USED

TIME-USED IS 1 MINS 5 SECS OUT OF 114 MINS 20 SECS

@

**@UNFIX**

See also UNFIX (MON 116). See also related @FIX and @FIXC.

**Function:**

Permit a segment to be swapped out after @FIX or @FIXC.

**Format:**

@UNFIX <segment no.>

**Parameters:**

<segment no.>  
(OCT).

**Rules:**

Permitted only for users RT and SYSTEM.

**Example:**

@UNFIX 35

Segment 35<sub>s</sub> can be swapped out.

**@UPDAT**

See also UPDAT (MON 111) and @CLADJ.

**Function:**

Give new values to the clock and calendar.

**Format:**

@UPDAT <minute> , <hour> , <day> , <month> , <year>

**Parameters:**

<minute>  
current minute (DEC:0-59, DEF = 0).  
<hour>  
current hour (DEC:0-23, DEF = 0).  
<day>  
current day (DEC:1-31)  
<month>  
current month (DEC:1-12)  
<year>  
current year (DEC, 4 digits)

**Rules:**

1. Permitted only for users RT and SYSTEM.
2. The time queue is unaffected by this command.
3. On NORD-100 the command updates the panel clock, if installed.

## @USER-STATISTICS

See also @LIST-USERS. See also related @LIST-FRIENDS.

### Function:

List the following for the user(s) for each directory on which they occur:

1. full name
2. date created
3. last date entered
4. default access for public, friend and owner
5. number of pages used
6. total number of pages

### Format:

@USER-STATISTICS [`<directory name>` :]`<user name>` ,`<output file>`

### Parameters:

[`<directory name>` :]`<user name>`

list data for all users matching this name. Can be abbreviated (DEF = all users in all default directories).

`<output file>`

destination of the list (DEF = TERMINAL).

### Rules:

Permitted for all users.

### Example:

@USER-STATISTICS SYSTEM, TERMINAL

USER 0: BIG-PACK: SYSTEM

CREATED 09. 31. 15 MARCH, 1978

LAST DATE ENTERED 16. 26. 14 JANUARY 19, 1979

DEFAULT PUBLIC ACCESS: READ

DEFAULT FRIEND ACCESS: READ, WRITE, APPEND, COMMON,  
DIRECTORY

DEFAULT OWN ACCESS: READ, WRITE, APPEND, COMMON, DIRECTORY

10505 PAGES USED OUT OF 11861 PAGES

## @WAIT-FOR-OPERATOR

See also related @OPERATOR and @RESTART-USER.

### Function:

Wait for the operator to restart the user (@RESTART-USER).

### Format:

@WAIT-FOR-OPERATOR

### Parameters:

None.

### Rules:

1. Permitted for all users.
2. The error device receives the message:  
 — — — hh.mm.ss WAITING TERMINAL ttt  
 where hh.mm.ss is the current time and ttt is the waiting terminal.

## @WFILE

See also WFILE (MON 120). See also related @RFILE, RFILE (MON 117) and @SET-BLOCK-SIZE.

### Function:

Transfer an area of the user's virtual memory to one or more random blocks in a file.

### Format:

@WFILE <file no.>, <memory address>, <block no.>, <no. of words>

### Parameters:

- <file no.>  
 logical device number from the @OPEN-FILE or @CONNECT-FILE command (OCT).
- <memory address>  
 starting address of memory record location (OCT, DEF = 0).
- <block no.>  
 number of destination file block (OCT: 0 —, DEF = 0).  
 —1 = write next block.
- <no. of words>  
 length of the area (OCT).

### Rules:

1. Permitted for all users.
2. May be used on communication channels.
3. Default block size is 256<sub>10</sub>.

### Example:

@WFILE 101, 400, 0, 1000

Write 1000<sub>8</sub> words to block 0 of file 101<sub>8</sub> from memory address 400<sub>8</sub>.



**@WHERE-IS-FILE**

See also WHDEV (MON 140). See also related @LIST-FILES, @LIST-OPENED-FILES and @LIST-RTOPEN.

**Function:**

Check whether a file is opened and/or reserved. List the user or RT program opening or reserving the file.

**Format:**

@WHERE-IS-FILE <file name>

**Parameters:**

<file name>

<object name> and <type> must be unique.

**Rules:**

1. Permitted for all users.
2. The possible messages are:
  - a) <file name> : OPEN BY USER <user name> ON TERMINAL <n>
  - b) <file name> : RESERVED BY RT-PROGRAM <program name>  
(A peripheral file is reserved but not opened.)
  - c) <file name> : FREE TO USE
  - d) <file name> : OPEN BY USER RT ON TERMINAL 1  
(if opened by an RT program).

**Example:**

@WHERE-IS (SCR)SCRATCH08:D

SCRATCH08: DATA: OPENED BY USER GUEST ON TERMINAL 39

④

## @WHO-IS-ON

See also related @TERMINAL-STATUS.

**Function:**

List logical device number and name of users logged in.

**Format:**

@WHO-IS-ON

**Parameters:**

None.

**Rules:**

Permitted for all users.

**Example:**

@WHO-IS-ON

39 GUEST

50 SYSTEM

@

User GUEST is logged in on terminal 39 and user SYSTEM on terminal 50. The arrow indicates the terminal which gave the command.

## 2.3 MONITOR CALLS IN ALPHABETICAL ORDER

### ABORT (MON 105)

See also @ABORT. See also related RTEXT.

**Function:**

Stop an RT program by setting it in the passive state. It is removed from the time or execution queue, all resources are released and periodic execution is prevented.

**Monitor call format:**

	LDA	(PAR	% A = address of parameter list
	MON	105	% ABORT
		...	
PAR,		(RTPRO	

**Input Parameters:**

RTPRO = address of RT description. 0 = the user's program.

**Output Parameters:**

None.

**Rules:**

1. Callable only from foreground.
2. Effective only if the program is running or waiting.

**Example:**

In Fortran:

EXTERNAL KLOKK

...

CALL ABORT (KLOKK)

If valid, the RT program KLOKK is aborted.

## ABSET (MON 102)

See also @ABSET. See also related @SET.

### Function:

Start an RT program at a specific time of day. The program is put in the time queue and moved to the execution queue at the specified time.

### Monitor call format:

	LDA	(PAR	% A = address of parameter list
	MON	102	% ABSET
	...		
PAR,	(RTPRO		
	ISEC		
	IMIN		
	IHOUR		

### Input Parameters:

RTPRO = address of RT description. 0 = user's program.

ISEC, IMIN, IHOUR = time of day when the program is put in the execution queue.

### Output Parameters:

None.

### Rules:

1. Callable only from foreground.
2. If the time of day has passed, the program is scheduled for the next day.
3. If the program is already in the time queue, it is removed and reinserted according to the new specifications.
4. Calling CLADJ to correct the internal clock may affect scheduling. RT programs scheduled by ABSET are then rescheduled according to the new time.

### Example:

In Fortran:

```
EXTERNAL RTPRO
```

```
...
```

```
CALL ABSET(RTPRO, 0, 30, 23)
```

Schedule RTPRO for execution at 23.30.

## ABSTR (MON 131)

See also related RFILE and WFILE.

### Function:

Transfer data between a physical memory area and a mass-storage device.

### Monitor call format:

LDT	LDN	% T = logical device no.
LDA	(PAR	% A = address of parameter list
MON	131	% ABSTR
JAN	ERROR	% Error if A < 0
...		
PAR,	IFUNC	
	DMEM	
	IBLOCK	
	NBLOCK	

### Input Parameters:

T = logical device number.

IFUNC = function code.\*

Bits 6-8: mass storage unit number (0-3)

Bits 0-5: function code

= 0: read

= 1: write

= 2: read test

= 3: compare

= 20: read status

\* See appropriate mass storage driver documentation.

DMEM = double integer memory address for data transfer.

IBLOCK = mass storage block address for data transfer. For cartridge disks:

bit 15 = 0; removable pack

bit 15 = 1; fixed pack.

For magnetic tape, this parameter contains the mass-storage unit number. Refer to the appropriate hardware manual for hardware block size.

NBLOCK = number of hardware blocks to be transferred. For magnetic tape, this parameter contains the record size (number of words).

### Output Parameters:

A ≥ 0: OK.

A < 0: Error

(bit 14-0 is hardware status)

### Rules:

1. Callable only from foreground.
2. Monitor call and parameters must be in permanent memory or on a fixed segment, residing on protection ring 2 page table 0.
3. The memory data area must be contiguous. When using older versions of magnetic tapes or disk controllers the transfer must not cross physical memory bank boundaries (128K bytes). Older versions have ND numbers less than ND-537 ( tape) or ND-559 (disk).

**Example:**

In Fortran:

INTEGER ABSTR

DOUBLE INTEGER MEMO

MEMO = 140000B

IX = ABSTR (502B, 0, MEMO, 20000B, 3)

Read 600<sub>8</sub> words from disk unit 0, block address 20000<sub>8</sub> to the physical memory address 140000<sub>8</sub>.

## ACM (MON 145)

**Function:**

Memory to memory transfer of data between two computers connected via a DMA channel.

**Monitor call format:**

	LDA	(PAR	% A = parameter list
	MON	145	% ACM
	STA	ISTAT	% return status
	...		
PAR,	LDN		
	IFUNC		
	MARRY		
	MDMA		
	NWORD		

**Input Parameters:**

LDN = logical device number of DMA channel

IFUNC = function code:

- = 0: read
- = 1: write
- = 2: lock/write/unlock
- = 3: clear

MARRY = address of data area in own computer.

MDMA = address of data destination in the other computer.

NWORD = number of words to transfer.

**Output Parameters:**

ISTAT = 0: OK

ISTAT  $\neq$  0: error number

**Rules:**

1. Callable from background and foreground.
2. An option which can be included at system generation time.

**Example:**

In Fortran:

INTEGER ACM

DIMENSION MARRY(N)

...

ISTAT = ACM(IUNIT, IFUNC, MARRY, MDMA, NWORD)

## **AIRDW (MON 37)**

**Function:**

Read a set of analog to digital channels (ND-820 AD convertor cards). See "NORD Process I/O Software Guide" ND-60.093 for further details.

## **ALTOF (MON 34)**

See also related ALTON.

**Function:**

Set the alternative page table (PT) equal to the normal page table.

**Monitor call format:**

MON 34

**Input Parameters:**

None.

**Output Parameters:**

None.

**Rules:**

Callable from background and foreground.

**ALTON (MON 33)**

See also related ALTOF.

**Function:**

Turn on/specify alternative page table (PT) to be used for X and B relative data references. (See NORD-10/100 Reference Manuals.)

**Monitor call format:**

LDA	(PAR	% A = address of parameter list
MON	33	% ALTON
...		% normal return. X and B relative
		% addressing are now through the
		% alternative PT
PAR,	IPIT	% address to parameter

**Input Parameters:**

IPIT = number of the alternative page table (must be = 3 for background programs, otherwise, 0, 1, 2, or 3).

**Output Parameters:**

None.

**Rules:**

1. Callable from background and foreground.
2. When used from background, the background segment size must be 128K. See @CHANGE-BACKGROUND-SEGMENT-SIZE.
3. ALTON must not be called twice without calling ALTOF in between.
4. If addressing via alternative page table (LDA, STA) status bit 0 should be set to 1. If this status bit = 0, addressing will be via normal page table.
5. Monitor call is independent of condition of status bit 0 collect/return parameters to alternative page table. (DEF alternative page table = normal page table.)
6. After ALTON all non-P-relative addresses go through the page table specified in the ALTON call.

**Example:**

In MAC:

LDA	(PAR	% A = address of parameter list
MON	33	% ALTON
...		% normal return
PAR,	(3	% use alternative PT = 3

In Fortran:

CALL ALTON(IPIT)



## ASSIG (MON 154)

### Function:

Assigns a graded LAM in the CAMAC ident table to a logical device number.  
See "NORD Process I/O Software Guide", ND-60.093.

### Monitor call format:

MON 154

### Input Parameters:

T = graded LAM (T = 0 is used for high priority on level 13)

A = logical device number.

D = crate number.

### Output Parameters:

A = 0: OK

A = -1: error in parameters

### Rules:

1. Callable only from foreground.
2. The call will disassign by specifying LDN = -1 (ie. A-register).

### Example:

In Fortran:

```
C LDN LOGICAL DEVICE NO.
C IGL GRADED LAM
C IC CRATE NUMBER
CALL ASSIG(LDN, IGL, IC)
```

**B4INW (MON 63)**

See also B8INB, M8INB and INBT. See also related B8OUT.

**Function:**

Read 8 bytes from a device.

**Monitor call format:**

MON 63

**Input Parameters:**

T = logical device number (do not use terminal).

**Output Parameters:**

Error return: A = error number

Skip return: A, D, L, X = bytes read (or words, if word oriented device)

T = 10<sub>8</sub>

**Rules:**

1. Callable from background and foreground.
2. See rule 2 of B8INB.

**B8INB (MON 23)**

See also B4INW, M8INB and INBT. See also related B8OUT.

**Function:**

Read up to 8 bytes from a device. The defined echo and break strategies do not apply.

**Monitor call format:**

MON 23

**Input Parameters:**

T = logical device number (do not use terminal)

**Output Parameters:**

Error return: A = error number

Skip return: A, D, L, X = bytes read

T = number of bytes read or number of words if word oriented device.

**Rules:**

1. Callable from background and foreground.
2. If the logical device uses a standard terminal driver, only the least significant 7 bits of the byte contain data. (The user should set the most significant bit of every byte to zero.)

## **B8OUT      (MON 24)**

See also M8OUT and OUTBT. See also related B8INB.

**Function:**

Write 8 bytes to a device.

**Monitor call format:**

MON 24

**Input Parameters:**

T = logical device number

A, D, L, X = bytes to be written.

**Output Parameters:**

Error return: A = error number

Skip return: OK

**Rules:**

Callable from background and foreground.

## BRKM (MON 4)

See also related ECHOM.

### Function:

Set break strategy for a terminal. Break strategies are described in SINTRAN III System Documentation, ND-60.062.

### Monitor call format:

MON 4

### Input Parameters:

T = logical device number (only for RT programs, otherwise break strategy for the user's terminal is set)

A < 0: no break

= 0: break on all characters

= 1: break only on control characters

= 2: special MAC break strategy

= 3 to 6: system defined break strategy

= 7: user defined break strategy. X register points to an 8 word bit map. The bit is set if the corresponding character should break (for example, bit 4 in the first word is set if character 4 should break).

D = maximum number of characters on input before break is given. Only if A ≥ 3.

### Output Parameters:

None.

### Rules:

1. Callable from background and foreground.
2. Break Table 7 is an option when SINTRAN is generated.
3. Parameters are fetched and returned to the user via the alternative page table.

### Example:

RT program in MAC:

LDT	LDN	% T = logical device number
SAA	1	% A = break on all control characters
MON	4	% BRKM
...		% normal return

Background program in Fortran (ND-100):

```
INTEGER ARRAY IARR(8)
```

```
...
```

```
CALL BRKM(1, 7, MAXCHR, IARR)
```

## CAMAC (MON 147)

### Function:

Operate the CAMAC (execute NAF). See also "NORD Process I/O Software Guide", ND-60.093.

### Monitor call format:

MON 147

### Input Parameters:

T = data if write

A = NAF

Bits 13 - 9: station number

Bits 8 - 5: subaddress

Bits 4 - 0: function

D = crate number in bit 9 - 6

### Output Parameters:

T = data if read

X = status

### Rules:

Callable only from foreground.

## CIBUF (MON 13)

See also @CLEAR-DEVICE. See also related COBUF and IOSET.

### Function:

Clear device input buffer.

### Monitor call format:

MON 13

### Input Parameters:

T = logical device number

### Output Parameters:

Error return: A = error number

Skip return: A = original contents

### Rules:

1. Callable from background and foreground.
2. Logical device number must not be 1 for background programs, ie. the user's terminal cannot be specified. Use RSIO (MON 143) to get the right logical device number.

### Example:

In Fortran:

```
C IUNIT  FORTRAN IO UNIT  10
      CALL CIBUF(IUNIT)
      IF(ERRCODE.NE.0) CALL ERROR
```

## CLADJ (MON 112)

See also related UPDAT, CLOCK and TIME.

### Function:

Set the clock forward or back by a specified amount.

### Monitor call format:

	LDA	(PAR	% A = address of parameter list
	MON	112	% CLADJ
PAR,	NOTIM		
	ITUN		

### Input Parameters:

NOTIM = number of time units the clock will be adjusted  
         < 0: the clock will stand still for NOTIM time units  
         > 0: the clock will be incremented NOTIM time units

ITUN = 1: basic time units (usually 20 msec)  
        = 2: seconds  
        = 3: minutes  
        = 4: hours

### Output Parameters:

None.

### Rules:

1. Callable only from foreground.
2. The scheduling time for RT programs inserted in the time queue by ABSET is modified. Other scheduling times are not affected. The next execution of a periodic program is changed if it was first inserted in the time queue by ABSET but subsequent intervals are unchanged.
3. On ND-100 with a panel clock, the clock will be adjusted.

### Example:

In Fortran:

```
CALL CLADJ (-15, 2)
```

The clock stands still for 15 seconds.

## CLOCK (MON 113)

See also @DATCL and TIME. See also related CLADJ and UPDAT.

### Function:

Read the current setting of clock and date.

### Monitor call format:

	LDA	(PAR
	MON	113
	...	
PAR,	CLDAT	
	...	
CLDAT,	0	% basic units
	0	% seconds
	0	% minutes
	0	% hours
	0	% day
	0	% month
	0	% year

### Input Parameters:

None.

### Output Parameters:

CLDAT = array as shown above.

### Rules:

1. Callable from background and foreground.
2. Parameters are fetched and returned to the user via the alternative page table.

### Example:

In Fortran:

```
INTEGER CLDAT(7)
CALL CLOCK(CLDAT)
```

## CLOSE (MON 43)

See also SPCLO and @CLOSE-FILE. See also related OPEN.

### Function:

Close one or more files.

### Monitor call format:

MON 43

### Input Parameters:

T > 0: logical device number of file to be closed.  
       = -1: close all user files not permanently open.  
       = -2: close all user files.

### Output Parameters:

Error return: A = error number  
 Skip return: A = original contents

### Rules:

1. Callable from background and foreground.
2. Close also resets peripheral files. This is equivalent to calling IOSET with KONTR = -1.

### Example:

In MAC:

SAT	-1	% close all files not permanently opened
MON	43	% CLOSE
JMP	ERROR	% error return
...		% normal return

## COBUF (MON 14)

See also @CLEAR-DEVICE. See also related CIBUF and IOSET.

### Function:

Clear device output buffer.

### Monitor call format:

MON 14

### Input Parameters:

T = logical device number.

### Output Parameters:

Error return: A = error number  
 Skip return: A = original contents

### Rules:

1. Callable from background and foreground.
2. See rule 2 of CIBUF.
3. If internal device is specified, then a break condition is generated starting the program on the receiving side, if any.



## COMND (MON 70)

See also related SETCM.

### Function:

Execute a SINTRAN command from a user program.

### Monitor call format:

MON 70

### Input Parameters:

A = address of command character string, which must be terminated by a single apostrophe (').

### Output Parameters:

The program is aborted if the command is in error.

### Rules:

1. Callable only from background.
2. If the contents of the user's virtual memory are destroyed by execution of the command return to the user's program may be impossible.
3. Parameters are fetched and returned to the user via the alternative page table.

### Example:

In Fortran:

```
CHARACTER DELFI * 50
DATA DELFI/'DELETE-FILE XXX:SYMB'/' /
CALL COMND (DELF)
```

The SINTRAN command DELETE-FILE XXX:SYMB is executed from a FORTRAN background program.

## CONCT (MON 106)

See also @CONCT. See also related DSCNT.

### Function:

Connect an RT program to the interrupt from a device. The RT program is put in the execution queue when an interrupt occurs from the device.

### Monitor call format:

	LDA	(PAR	% A = address of parameter list
	MON	106	% CONCT
	...		
PAR,	(RTPRO		
	LDN		

### Input Parameters:

RTPRO = address of RT description. 0 = calling program.

LDN = logical device number.

### Rules:

1. Callable only from foreground.
2. Several units may be connected to one program.
3. Valid only if the device has a "connect" driver routine.

### Example:

In Fortran:

EXTERNAL KLOKK

...

CALL CONCT(KLOKK,401B)

The RT description of KLOKK will be put in the execution queue whenever device 401<sub>s</sub> gives an interrupt.

## CRALF (MON 221)

### Function:

Create or allocate file from monitor call. User SYSTEM/RT can create/allocate files in any user's file space. Public users must have directory access to create/allocate file. If start-address is zero then create-file, if nonzero then allocate-file. When create-file, if the number of pages = 0 the file will be indexed, otherwise it will be contiguous.

### Monitor call format:

MON 221

### Input parameters:

X = memory address of source user string, ie. file name  
 A&D = start-address, ie. page address on disk of first page of file  
 T = address of double word with number of pages.

### Output parameters:

Error exit: A = error number

Skip return: OK

## DABST (MON 127)

See also ABSET.

### Function:

Start an RT program at a specific time given in basic time units.

### Monitor call format:

```

          LDA      (PAR      % A = address of parameters list
          MON      127      % DABST
          ...
PAR,      (RTPRO
          DTIM

```

### Input Parameters:

RTPRO = RT program address. 0 = user's program

DTIM = double word containing the time when the program is put in the execution queue. The number is in basic time units.

### Output Parameters:

None.

### Rules:

1. Callable only from foreground.
2. See rules 2 to 4 of ABSET.
3. Current time in basic time units can be read by TIME (MON 11).

### Example:

In Fortran:

```
EXTERNAL KLOKK
```

```
DOUBLE INTEGER DTIM
```

```
...
```

```
DTIM = 3000000
```

```
CALL DABST(KLOKK, DTIM)
```

KLOKK is started when time reaches 3000000 basic time units.

In MAC:

```

          )9EXT      KLOKK
          ...
          LDA      (PAR
          MON      127
          ...
PAR,      (KLOKK
          1                      % DTIM, most significant word
          100000             % DTIM, least significant word

```

## DESCF (MON 71)

See also @DISABLE-ESCAPE-FUNCTION. See also related EESCF (MON 72).

**Function:**

Disable escape function. I.e. no "user break" is performed until the next EESCF (MON 72).

**Monitor call format:**

MON 71

**Input Parameters:**

T = logical device number (foreground programs; for background programs, the user's terminal is assumed and T register is ignored).

**Output Parameters:**

None.

**Rules:**

Callable from background and foreground.

**Example:**

In Fortran:

```
CALL DESCF(LDN)
```

**DINTV (MON 130)**

See also INTV.

**Function:**

Prepare an RT program for periodic execution. The period is specified in basic time units.

**Monitor call format:**

	LDA	(PAR	% A = address of parameter list
	MON	130	% DINTV
		...	
PAR,	(RTPRO		
	DTIM		

**Input Parameters:**

RTPRO = address of RT description. 0 = calling program.

DTIM = double integer containing the interval between the times when the program is put in the execution queue. The number is in basic time units.

**Rules:**

1. Callable only from foreground.
2. See rules 2 - 6 of INTV.

**Example:**

In Fortran:

EXTERNAL KLOKK

DOUBLE INTEGER DTIM

...

DTIM = 300000B

CALL DINTV(KLOKK,DTIM)

The RT program KLOKK is put in the execution queue every 300000<sub>B</sub> basic time units, starting at the next call.

In MAC:

)9EXT KLOKK

...

LDA (PAR

MON 130

...

PAR, (KLOKK

1

% DTIM most significant word

100000

% DTIM least significant word

**DIW (MON 165)**

See also related DOLW.

**Function:**

Read the current digital status for a group of process input registers. See "NORD Process I/O Software Guide", ND-60.093.

**DMAC BREAKPOINT (MON 51)**

See also SBRK.

**Function:**

This is a special monitor call used by the DMAC assembler for setting breakpoints. Not available for other users.

**DOLW (MON 166)**

See also related DIW.

**Function:**

Write to a set of process output registers. See "NORD Process I/O Software Guide" ND-60.093.

**DOPEN (MON 220)****Function:**

Open a file for read and/or write access. Identical with OPEN (MON 50) except when used by user SYSTEM or RT, when DOPEN bypasses normal file access and uses owner access rights.

**Monitor call format:**

See OPEN (MON 50).

## DROBJ (MON 215)

### Function:

Read object entry specified by directory, user and object index.

### Monitor call format:

MON 215

### Input parameters:

A = memory address of destination object entry buffer (min 40 words)

T = directory index (left byte) and user index (right byte)

X = object index

### Output parameters:

Error return: A = error number

Skip return: OK

### Rules:

1. User SYSTEM may read any object entry.
2. Public users may only read the object entries of files to which they have read access.

## DSCNT (MON 107)

See also @DSCNT. See also related CONCT and INTV.

### Function:

Remove all connections to an RT program made by CONCT (or @CONCT). If the program is in the time queue it is removed and periodic execution is prevented.

### Monitor call format:

LDA	(PAR	% A = address of parameter list
MON	107	%DSCNT
	...	
PAR,	(RTPRO	

### Input Parameters:

RTPRO = address of RT description 0 = calling program.

### Output Parameters:

None.

### Rules:

1. Callable only from foreground.
2. Reserved resources are not released.
3. The program is not removed from the execution queue.

### Example:

In Fortran:

CALL DSCNT(RTPRO)

## DSET (MON 126)

See also SET.

### Function:

Enter an RT program in the time queue and transfer it to the execution queue after a specified number of basic time units.

### Monitor call format:

	LDA	(PAR	% A = address of parameter list
	MON	126	% DSET
		...	
PAR,	(RTPRO		
	DTIM		

## DWOBJ (MON 216)

### Function:

Write access, last date opened for read, last date opened for write, max. byte pointer specified by directory, user and object index. No file pointers are affected. User SYSTEM can write any object entry providing the file is closed and object entry source filename exactly matches destination object entry. Public users can only write their own object entries.

### Monitor call format:

MON 216

### Input parameters:

A = memory address of source object entry buffer (min 40<sub>8</sub> words)

T = directory index (left byte) and user index (right byte)

X = object index

### Output parameters:

Error return: A = error number

Skip return: OK

### Rules:

1. The file must be closed; source file name and destination file name must match exactly.
2. User SYSTEM may write any object entry.
3. Public users may only write their own object entries.



## ECHOM (MON 3)

See also related BRKM.

### Function:

Set echo strategy for a terminal. Description of echo strategies is found in "SINTRAN III System Documentation", ND-60.062.

### Monitor call format:

MON 3

### Input Parameters:

T = logical device number (foreground programs; for background programs, the user's terminal is assumed and T register contents ignored.)

A < 0: no echo.

A = 0: echo on all characters

A = 1: echo on all except control characters

A = 2: special MAC echo strategy

A = 3 to 6: system defined echo strategy

A = 7: user defined echo strategy. X register points to an 8 word bit map. The bit is set if the corresponding character should give echo (for example, bit 4 in the first word is set if character 4 should give echo).

### Output Parameters:

None.

### Rules:

1. Callable from background and foreground.
2. Break Table 7 is an option to be defined at SYSTEM generation time.

### Example:

In Fortran:

```
INTEGER ARRAY IARRY(8)
```

```
...
```

```
CALL ECHOM(LDN, ISTR, IARRY)
```

RT Program in MAC:

LDT	LDN	% T = logical device number
SAA	3	% A = system strategy number 3
MON	3	% ECHOM
...		% normal return

**EDTRM (MON 206)**

See also @ENABLE-TERMINATION-HANDLING and  
@DISABLE-TERMINATION-HANDLING.

**Function:**

Enable/disable termination handling of background and RT programs.

**Monitor call format:**

	LDA	(PARLI	% address of parameter
	MON	206	% list in A-register
	...		
PARLI,	EDFLA		
	RTUSF		
EDFLA,	0		% enable (= 1), disable (= 0) flag
RTUSF,	0		% RT, user break and/or fatal error flag
			% (0,1,2 or 3)

**Rules:**

1. Callable from background and foreground.
2. For RT programs termination handling is either on or off.
3. For background programs termination handling can be on or off for either user break, fatal error or both.

**EESCF (MON 72)**

See also @ENABLE-ESCAPE-FUNCTION. See also related DESCF  
(MON 71).

**Function:**

Enable the "user break" function.

**Monitor call format:**

MON 72

**Input Parameters:**

T = logical device number (foreground programs; for background programs, the user's terminal is assumed and T register is ignored.)

**Output Parameters:**

None.

**Rules:**

Callable from background and foreground.

**Example:**

In Fortran:

CALL EESCF(LDN)

## ENTSG (MON 157)

See also @ENTSG.

### Function:

Enter a routine as a direct task or as a device driver.

### Monitor call format:

	LDA	(PAR	% A = address of parameter list
	MON	157	% ENTSG
	...		
PAR,	ISEG		
	IPT		
	INTLEV		
	ISTART		

### Input Parameters:

ISEG = the segment where the routine resides.

IPT = the page table to be used for the segment.

INTLEV = the interrupt level under which the program is to run (DEC). Must not be 10, 11, 12 or 13. If a device driver routine is to be entered on one of these levels, a free level (2, 5, 6, 7, 8 or 9) must be specified instead.

ISTART = entry point of direct task or device driver routine. Value is a virtual address.

### Output Parameters:

None.

### Rules:

Callable only from foreground.

### Example:

In Fortran:

```
CALL ENTSG (36B, 3, 6, 50000B)
```

A routine loaded on segment 36<sub>h</sub> is entered as a direct task using page table 3, interrupt level 6. The routine starts on address 50000<sub>h</sub>.

**ERMON (MON 142)**

See also ERMSG. See also related MSG.

**Function:**

Write a user defined run-time error on the error device.

**Monitor call format:**

LDA	IERR	% A = error no. as 2 ASCII char. in the % range 50-69
LDT	ISUB	% T = suberror number
MON	142	% ERMON
...		% normal return

**Input Parameters:**

IERR = error number, as two ASCII characters in the range 50 - 69. Parity bits are cleared.

ISUB = suberror number as a positive integer.

**Output Parameters:**

None.

**Rules:**

1. Callable only from foreground.
2. The error message is printed as follows:  
hh.mm.ss ERROR ierr IN rtprog AT addr; USER ERROR  
SUBERROR: isub  
where  
nn.mm.ss is the time when the error message was printed.  
hh - hours  
mm - minutes  
ss - seconds  
ierr is error number.  
rtprog is RT program name.  
addr is octal address (virtual address where the error occurred.)  
isub is suberror number.

**Example:**

In Fortran:

Current RT program is RTPA.

CALL ERMON(2H59, 100)

This means that the following error message will be printed on the error device:

15. 45. 02 ERROR 59 IN RTPA AT 572; USER ERROR.

SUBERROR: 100

In MAC:

...		
LDA	(59	% error no. = 59
SAT	100	% suberror no. = 100
MON	142	% ERMON
...		

## ERMSG (MON 64)

See also QERMS.

### Function:

Print the error message for a given system error number. The program does not terminate.

### Monitor call format:

MON 64

### Input Parameters:

A = error number of error message to be printed.

### Output Parameters:

None.

### Rules:

1. Callable from background and foreground.
2. The message is output to the user terminal, except when the program is running under a batch or mode file or running as an RT program. Then the message is output to the error device.

## EXIOX (MON 31)

See also @EXECUTE-IOX and IOXN. See also related IPRIV.

### Function:

Execute an IOX instruction.

### Monitor call format:

LDA	(PAR	% A = address of parameter list
MON	31	% EXIOX
STA	IAOUT	% contents of A register after the IOX
...		
PAR,	IAIN	
	IDRA	

### Input Parameters:

IAIN = contents of A register before execution.

IDRA = device register address, bits 10 - 0 of the IOX instruction.

### Output Parameters:

IAIN = function value.

IAOUT = contents of A register after execution (status).

### Rules:

Only device register addresses known to SINTRAN are accepted.

## **FIX (MON 115)**

See also @FIX and @FIXC. See also related UNFIX.

### **Function:**

Fix a nondemand segment in memory, ie. after the call it will no longer be allowed to be swapped out. This function is useful for certain time critical programs or allocating DMA buffers. The pages could be scattered "randomly" in memory.

### **Monitor call format:**

LDA	(PAR	% A = address of parameter list
MON	115	% FIX
...		% normal return
PAR,	ISEG	

### **Input Parameters:**

ISEG = segment number to be fixed. Must be a nondemand segment.

### **Output Parameters:**

None.

### **Rules:**

1. Callable only from foreground.
2. If ISEG refers to a nonexistent or demand segment, an error message is output and the program aborted.
3. The total number of pages that can be used for fixed segments is a system parameter. It can be specified at system generation time or through the command \*CHANGE-VARIABLE in @SINTRAN-SERVICE-PROGRAM.
4. UNFIX must be called or @UNFIX given before the RT loader can clear the segment.

### **Example:**

In Fortran:

```
CALL FIX(ISEG)
```

## FIXC (MON 160)

See also FIX and @FIXC. See also related UNFIX.

### Function:

Fix a segment in a contiguous area of physical memory.

### Monitor call format:

LDA	(PAR	% A = address of parameter list
MON	160	% FIXC
STA	ERCOD	% return
...		
PAR,	ISEG	
	IPAGE	

### Input Parameters:

ISEG = segment number to be fixed.

IPAGE = first physical page number to be loaded.

### Output Parameters:

Return status (only if bit 15 is set in segment number parameter).

A = 0 OK

- = -1 space not available
- = -2 illegal segment
- = -3 segment not loaded
- = -4 attempt to fix demand segment
- = -5 attempt to fix too many pages
- = -6 segment already fixed at a different address

### Rules:

1. Callable only from foreground.
2. UNFIX must be called or @UNFIX given before the RT loader can clear the segment.
3. See FIX rules 2 and 3.
4. If an error condition occurs, a runtime error message is output on the error device and the program aborted, but if the most significant bit is set in the segment number parameter (ISEG) a return status is given to the calling program A-register.

### Example:

In Fortran:

```
CALL FIXC(77B, 170B)
```

The segment number 77<sub>8</sub> is fixed in memory starting at physical address 360000<sub>8</sub> (page number 170B).

## FIXC5 (MON 61)

### Function:

A special form of FIXC in SINTRAN III VSE/500 for the ND-500 monitor to allocate memory. Some functions are available in SINTRAN III VSE for user programs. It can be used to fix a contiguous area of physical memory within a given memory range, or allocate a given number of pages (not segments) within a specified range. This is useful for allocation of large, contiguous DMA buffers.

### Monitor call format:

```

          LDA      (PARLI
          MON      61
          ...
PARLI,    PAR1, P1
          PAR2, P2
          PAR3, P3
          PAR4, P4
          PAR5, P5
          PAR6, P6

```

### Parameters:

There are 6 parameters. The first is always the function code, the meaning of the others depends on the function code.

Function codes 1, 2 and 3 are reserved for ND-500 monitor and must never be used by any other programs.

Function code 4: Fix a segment contiguously at any address within a specified area.

```

PAR1 = 4 (function code)
PAR2 = segment number (octal)
PAR3 = first (lower) legal physical page
PAR4 = last (upper) legal physical page
PAR5 = not used
PAR6 = not used

```

```

Error return: A = 0 area occupied
               = 1 PAR3 > last physical page in memory
               = 2 something wrong with the segment
                  (demand, already fixed etc.)

```

```

Skip return: OK. Segment is fixed contiguously.
              A = first physical page in segment.

```

Function code 5: Reserve a contiguous memory area.

```

PAR1 = 5 function code
PAR2, PAR3 see function code 4
PAR5 = not used
PAR6 = not used

```

```

Error return: A = 0 area occupied
               = 1 PAR3 > last physical page in memory
               = 2 no free table element in table for allocated memory areas
                  (system generation)

```

```

Skip return: OK. Area allocated.
              A = first physical page in area
              T = area index (to be used when deallocating area)
                  see function code 6

```



Function code 6: Release memory area reserved by function code 5.

PAR1 = 6 function code

PAR2 = table index (T-reg on return from function 5)

PAR3 = first physical page in area (A-reg on return from function 5)

PAR4 = not used

PAR5 = not used

PAR6 = not used

Return: error

Skip return: OK

**Rules:**

1. Callable from background and foreground.
2. This monitor call should not normally be used by application programs but function codes 4, 5 and 6 can be used to perform certain operations.
3. Return status in A-register (if not skip return).

**Input Parameters:**

RTPRO = address of RT description. 0 = calling program.

DTIM = double integer containing the number of basic time units the program is to stay in the time queue.

**Output Parameters:**

None.

**Rules:**

1. Callable only from foreground.
2. See rules 2 - 4 of SET.

**Example:**

In Fortran:

EXTERNAL KLOKK

DOUBLE INTEGER DTIM

DTIM = 500

CALL DSET(KLOKK,DTIM)

The program KLOKK will be set up for execution 500 basic time units after the call.

In MAC:

)9EXT KLOKK

...

LDA (PAR

MON 126

...

PAR, (KLOKK

0

% DTIM, most significant word

500

% DTIM, least significant word

**GBSIZ (MON 222)****Function:**

Get the size of the background segment.

**Monitor call format:**

MON	222	% GBSIZ
STA	BSGSZ	% segment size in A-register

**Input Parameters:**

None.

**Output Parameters:**

A = 100<sub>8</sub> or 200<sub>8</sub> (64<sub>10</sub>K or 128<sub>10</sub>K, one or two memory banks).

**Rules:**

Callable from background only.

**GETRT (MON 30)**

See also GRTDA.

**Function:**

Get the RT description address of the calling program.

**Monitor call format:**

MON	30	% GETRT
STA	IRTAD	% A = output parameter

**Input Parameters:**

None.

**Output Parameters:**

A = address of RT description.

**Rules:**

Callable from background and foreground.

**Example:**

In Fortran:

INTEGER GETRT

...

IRTAD = GETRT(0)

C THE FORTRAN ROUTINE REQUIRES A DUMMY PARAMETER:

C IRTAD = ADDRESS OF RT DESCRIPTION

## GL (MON 150)

**Function:**

Read the CAMAC GL register or the last CAMAC ident number. See also "NORD Process I/O Software Guide", ND-60.093.

**Monitor call format:**

MON 150

**Input Parameters:**

A = function  
= -1: read last ident  
≠ -1: read GL register  
D = crate number in bits 9 - 6.

**Output Parameters:**

A = last ident or GL register.

**Rules:**

Callable only from foreground.

## GRAPHIC (MON 155)

### Function:

Execute graphic function for graphic NORDCOM system, pen plotter or Textronix display.

### Monitor call format:

	LDA	(PAR	% A = address of parameter list
	MON	155	% GRAPHIC
	STA	ISTAT	% A = return status
		...	
PAR,	IX		
	IY		
	IN		
	LDN		
	IFUNC		

### Input Parameters:

IX, IY = End coordinates of new line relative to current reference point.

IN = Integer code

LDN = logical device number.

IFUNC = routine select

= 0: Go PLOT

= 1: Go PLOTS (routine to establish reference point and/or clear a NORDCOM screen)

= 2: Go NEWP (routine to select pen or screen)

### Output Parameters:

ISTAT = output parameter for PLOTT.

### Rules:

1. Callable from background and foreground.
2. Some functions require a delay by the calling program. Refer to the specifications of the peripheral for the required timing.
3. The old call PLOTT is obsolete.

## GRTDA (MON 151)

See also GETRT. See also related GRTNA.

### Function:

Get the RT description address of an RT program.

### Monitor call format:

LDA	(PAR	% A = address of parameter list
MON	151	% GRTDA
STA	RTPRO	% A = address of RT description
...		

PAR, RTNAME

### Input Parameters:

RTNAME = RT program name as a character string terminated by a single apostrophe.

### Output Parameters:

A > 0: address of RT description.

A < 0: no RT program with this name.

### Output Parameters:

RTPRO > 0: address of RT description.

< 0: no RT program with this name.

### Rules:

Callable only from foreground.

### Example:

In Fortran:

```
INTEGER GRTDA, RTPRO
```

```
...
```

```
RTPRO = GRTDA(6HRTPGA')
```

```
IF (RTPRO.GT.0) CALL ABORT(RTPRO)
```

## GRTNA (MON 152)

See also @GET-RT-NAME. See also related GRTDA.

### Function:

Get the name of an RT program.

### Monitor call format:

LDA	(PAR	% A = address of parameter list
MON	152	% GRTNA
...		% T, A, D contain name
PAR,	RTDES	

### Input Parameters:

RTDES = address of RT description. 0 = calling program.

### Output Parameters:

D = 0: name not found.

D ≠ 0: T, A, D registers contain name. Right-justified character length is 6 bits.

### Rules:

Callable only from foreground.

## GUIOI (MON 217)

### Function:

Get directory, user and object indexes from open file number.

### Monitor call format:

MON 217

### Input Parameters:

A = open file number.

### Output Parameters:

Error return: A = error number.

Skip return: OK.

T = directory index (left byte) and user index (right byte).

X = object index.

## **GUSNA (MON 214)**

**Function:**

Get user-name.

**Monitor call format:**

MON 214

**Input Parameters:**

A = memory address to which user name should be written (16 characters).

X = directory index (left byte) and user index (right byte).

**Output Parameters:**

Error return: A = error number.

Skip return: OK.

## HOLD (MON 104)

See also @HOLD. See also related RTWT.

### Function:

Set the calling program in a waiting state for a specified period. The program is then continued after the call to HOLD.

### Monitor call format:

	LDA	(PAR	% A = address of parameter list
	MON	104	% HOLD
		...	
PAR,	NOTIM		
	ITUN		

### Input Parameters:

NOTIM	= number of time units the program will be in the wait state.
	= 0: clear 5REP bit in RT description without waiting for next clock interrupt (performance improvement).
ITUN	= 1: basic time units (usually 20 msec)
	= 2: seconds
	= 3: minutes
	= 4: hours

### Output Parameters:

None.

### Rules:

1. Callable from background and foreground.
2. No resources are released.
3. If no wait mode is set HOLD will exit when a break occurs. See NOWT (MON 36).
4. If the repeat flag is set there is an immediate return from the monitor call.
5. If <number of time units> = 0 this results in the repeat bit being reset and immediate return (without waiting until the next clock interrupt).

### Example:

In Fortran:

```
CALL HOLD(10,2)
```

The calling program waits for 10 seconds before continuing with the next statement.



**INBT (MON 1)**

See also M8INB and B8INB. See also related OUTBT.

**Function:**

Read one byte from a device. If the device is a data link or word oriented internal device, read one word.

**Monitor call format:**

MON 1

**Input Parameters:**

T = logical device number.

**Output Parameters:**

Error return: A = error number.

Skip return: A = data, see INCH below.

**Rules:**

1. Callable from background and foreground.
2. If there is no data in the input buffer, the program enters the I/O wait state. This can be changed by NOWT (MON 36).
3. When data is taken from a mass-storage file, the byte pointer is incremented by one (see SETB).
4. For card reader, the card column is converted to ASCII; the 12 bit card column can be read directly using IOSET.
5. If logical device number T = 0 for background program, then echo and break strategy is set to 1 (see ECHOM and BRKM) input is taken from the command input buffer and normal SINTRAN line editing can be used.
6. Reading from LDN 0 is reading from SINTRAN command buffer (background). SINTRAN line editing can be used. (Lower case characters are converted to upper case.)

**Example:**

In MAC:

LDT	FNUM	% T register = LDN
MON	1	% INBT
JMP	ERR	% error return, A = error number
STA	CHAR	% skip return, A = data
...		

In FORTRAN:

```
C IFNUM FORTRAN IO UNIT NUMBER
  ICHAR = INCH (IFNUM)
  IF (ERRCODE.NE.0) STOP
```

## INSTR (MON 161)

See also related OUTST.

### Function:

Read a string of characters from a peripheral device.

### Monitor call format:

	LDA	(PAR	% A = address of parameter list
	MON	161	% INSTR
	STA	ISTAT	% A = function value
	...		
PAR,	LDN		
	MARRY		
	MCHAR		
	ITERM		

### Input Parameters:

LDN = logical device number of a peripheral device.

MARRY = name of input data buffer.

MCHAR = maximum number of characters to be read.

ITERM = terminating character; input stops after this character is read.

### Output Parameters:

ISTAT = -1: error in parameters.  
 = 0 in bits 15, 14: maximum number of characters is read.  
 = 1 in bits 15, 14: a terminator is found.  
 = 2 in bits 15, 14: the device buffer is read without terminating on MCHAR or ITERM. ( RT programs only; emptying the buffer in background causes I/O wait.)  
 = 3 in bits 15, 14: device error. Bits 7 - 0 contain the error numbers. If bits 15 and 14 are 0 to 2, bits 13 - 0 contain the number of characters read.

### Rules:

1. Callable from background and foreground.
2. If the device buffer is empty, the calling program is set in a waiting state.
3. When called from RT there is immediate return if there are no characters in the internal input buffer.
4. Logical device number must not be 1 (own terminal). Use RSIO (MON 143) to get the appropriate logical device number.
5. Parameters are fetched and returned to the user via the alternative page table.

### Example:

In Fortran:

```

      INTEGER ITERM
      CHARACTER STRING * 100
      DATA ITERM/47B/
      MCHAR = 100
      ...
      ISTAT = INSTR(IUNIT, STRING, MCHAR, ITERM)
C   READ FROM IUNIT UNTIL AN ' IS READ OR 100 CHARACTERS HAVE BEEN READ
```

## INTV (MON 103)

See also @INTV.

### Function:

Prepare an RT program for periodic execution.

### Monitor call format:

	LDA	(PAR	% A = address of parameter list
	MON	103	% INTV
	...		
PAR,	(RTPRO		
	NOTIM		
	ITUN		

### Input Parameters:

RTPRO = address of RT description. 0 = calling program.

NOTIM = interval between each time the program is put in the execution queue.

ITUN = 1: basic time units (normally 20 msec).

= 2: seconds

= 3: minutes

= 4: hours

### Output Parameters:

None.

### Rules:

1. Callable only from foreground.
2. To start the first execution use RT or SET (or @RT or @SET).
3. Periodic execution is reset by a call to DSCNT or ABORT (or @DSCNT or @ABORT).
4. The interval can be changed at any time by another call to INTV (or @INTV).
5. If one execution is unfinished when it is time for the next execution, the program is scheduled for repeated execution. If the delay becomes as long as two intervals, one execution is lost.
6. Scheduling is unaffected by calling CLADJ.

### Example:

In Fortran:

```
EXTERNAL KLOKK
```

```
...
```

```
CALL INTV(KLOKK, 30, 2)
```

The RT program KLOKK is put in the execution queue every 30 seconds, starting at the next call to RT or SET.

## IOSET (MON 141)

See also @IOSET.

### Function:

Set control information for device.

### Monitor call format:

	LDA	(PAR	% A = address of parameter list
	MON	141	% IOSET
	MON	65	% error return
	...		% normal return
PAR,	LDN		
	IOF		
	IProg		
	KONTR		

### Input Parameters:

LDN = logical device number (not a file number).  
 IOF = 0: input part.  
       = 1: output part.  
 IProg = 0: the device is reserved by the user's terminal (background) or  
           by the calling RT program (foreground).  
       ≠ 0: address of an RT description. The device is assumed to be  
           reserved by this RT program. If the value of IProg is not  
           correct, a return is made with A register = -1.  
 KONTR = -1: reset device.  
 Parameter values for card reader are as follows:  
       -1: clear buffer and set ASCII mode — all characters are  
           converted to ASCII code, and trailing blanks are ignored.  
       0: set ASCII mode.  
       1: set binary mode; subsequent INBT monitor calls return a 12 bit  
           column image.

For some devices, eg. Process I/O and NORDCOM, the meaning of <KONTR>  
 is described elsewhere. (See also @IOSET.)

### Subroutine call format:

ISTAT = IOSET (LDN, IOF, IProg, KONTR).

### Input Parameters:

LDN = logical device number (must be a peripheral).  
 IOF, IProg, KONTR = see above.

### Output Parameters:

ISTAT = 0: OK.  
       ≠ 0: illegal logical unit or IProg does not match.

### Rules:

1. Callable from background and foreground.
2. Logical device number must not be 1 (own terminal). Use RSIO (MON 143) to get the logical number.
3. Logical device number must be reserved.
4. Control code definition is device dependent. See "Special I/O Guide" or SINTRAN listing (IOSET-continue) for details on an actual device.

**IOUT (MON 35)**

See also related MSG.

**Function:**

Print a number in octal or decimal format on the user's terminal.

**Monitor call format:**

MON 35

**Input Parameters:**

T = 18: octal format.

T = 10: decimal format.

A = number to be printed.

**Output Parameters:**

None.

**Rules:**

Callable only from background.

**Example:**

In MAC:

LDT	(12	% 10 = 12 <sub>8</sub>
LDA	(144	% 100 = 144 <sub>8</sub>
MON	35	

The number 100<sub>10</sub> is written on the terminal.

## IOXN (MON 153)

See also EXIOX.

### Function:

Execute a single IOX instruction for CAMAC. See also "NORD Process I/O Software Guide", ND-60.093.

### Monitor call format:

MON 153

### Input Parameters:

A = data if write.

D = hardware device number ( $2000_8 \leq D < 4000_8$ )

### Output Parameters:

A = data if read.

### Rules:

Callable only from foreground.

### Example:

In Fortran:

C ID - data if a write operation

C IC - IOX code

```
...
CALL IOXN(ID, IC)
```

**IPRIV (MON 146)**

See also related EXIOX.

**Function:**

Execute privileged instruction.

**Monitor call format:**

MON 146 % IPRIV

**Input Parameters:**

T = instruction to be executed.

**Output Parameters:**

None.

**Rules:**

1. Callable only from foreground.
2. The instruction uses the registers of the calling program.

**Example:**

In MAC:

```

LDT      (IOX 305 % T = instruction
SAA      A      % write ASCII 'A'
           % into the write data register of terminal
           1
MON      146     % IPRIV
```

## ISIZE (MON 66)

See also related OSIZE.

### Function:

Get the current number of bytes in the input buffer.

### Monitor call format:

MON 66

### Input Parameters:

T = logical device number.

### Output Parameters:

Error return: A = error number.

Skip return: A = number of bytes.

### Rules:

1. Callable from background and foreground.
2. Logical device number must not be 1 (own terminal). Use RSIO (MON 143) to get the logical device number.
3. If the logical device number is not reserved, the calling program is aborted.

### Example:

In MAC:

SAT	2	% T = tape reader
MON	66	% ISIZE
MON	65	% error return
STA	NUMB	% normal return

In Fortran:

```
OPEN(UNIT = IUNIT, ... )
NUMB = ISIZE(IUNIT)
IF(NUMB.LT.0) THEN CALL ERROR
ELSE
...
```



## LASTC (MON 26)

See also related INBT.

### Function:

Get the last character typed on a terminal.

### Monitor call format:

LDA	(PAR	% A = address of parameter list
MON	26	% LASTC
MON	65	% error return
STA	KAR	% A = character
...		
PAR,	LDN	

### Input Parameters:

LDN = logical device number of a terminal.

### Output Parameters:

Error return: A = error number.

Skip return: A = last typed character (8 bits)

### Rules:

1. Callable only from foreground.
2. The terminal need not be reserved by the calling program.

## LEAVE (MON 0)

See also RTEXT.

### Function:

Terminate the program being executed and return control to the operating system. Batch jobs are not terminated (see RTEXT, MON 134).

### Monitor call format:

MON 0

### Input Parameters:

None.

### Output Parameters:

None.

### Rules:

1. Callable from background and foreground.
2. If called from background all files not permanently opened are closed.
3. If called from foreground no files are closed but all reserved peripheral devices are released.
4. For batch jobs the program is terminated but the batch job continues (see RTEXT).

## M8INB (MON 21)

See also B8INB and INBT. See also related M8OUT.

**Function:**

Read up to 8 bytes from a device. The defined echo and break strategies apply.

**Monitor call format:**

MON 21

**Input Parameters:**

T = logical device number.

**Output Parameters:**

Error return: A = error number.

Skip return:

A, D, L, X = bytes read.

T = number of bytes read or number of words if word oriented device.

**Rules:**

1. Callable from background and foreground.
2. Can be used only on terminals, internal devices and NORDNET channels.

## M8OUT (MON 22)

See also B8OUT and OUTBT. See also related M8INB.

**Function:**

Write up to 8 bytes to a device.

**Monitor call format:**

MON 22

**Input Parameters:**

T = logical device number.

A, D, L, X = bytes to be written. A byte = 0 will not be written and terminates the output.

**Output Parameters:**

Error return: A = error number.

Skip return: OK.

**Rules:**

1. Callable from background and foreground.
2. Can be used only on terminals, internal devices and NORDNET channels.

## MAGTP (MON 144)

See also @DEVICE-FUNCTION.

### Function:

Perform operations on magnetic tapes, cassette tapes, Versatec printer/plotter, floppy disk and NORDNET channels.

### Monitor call format:

	LDA	(PAR	% A = parameter list
	MON	144	% MAGTP
	STA	ISTAT	% A register = function value
	...		
PAR,	IFUNC		
	MARRY		
	LDN		
	IPAR1		
	IPAR2		

### Input Parameters:

MARRY = memory area used for data transfer.

LDN = logical device number.

IPAR1 and IPAR2 = device-dependent parameters. See @DEVICE-FUNCTION.

IFUNC = function to be performed, see table below (OCT).

IFUNC	FUNCTION
0	READ-RECORD
1	WRITE-RECORD
2	READ-ODD-NUMBER-OF-BYTES
3	LOOP-WRITE-TO-READ-IN-FCU
4	READ-ONE-RECORD-BACKWARDS
5	UNLOCK-AND-STOP
6	LOCK-CASSETTE
7	ERASE-TAPE
10	ADVANCE-TO-EOF
11	REVERSE-TO-EOF
12	WRITE-EOF
13	REWIND
14	WRITE-ERASE-GAP
15	BACKSPACE-RECORDS
16	ADVANCE-RECORDS
17	UNLOAD
20	READ-STATUS
21	CLEAR-DEVICE
22	CLEAR-DEVICE (ERROR EXIT)
23	SELECT-DENSITY/PARITY-AND-DENSITY
24	READ-LAST-STATUS
25	READ-TAPE-STATUS
26	READ-BYTE-RECORD
27	WRITE-BYTE-RECORD
30	SET-ALPHANUMERIC-MODE
31	SET-GRAPHIC-MODE
32	GIVE-FORM-FEED
33	CLEAR-SELECTED-UNIT
34	SET-DIAGNOSTIC-MODE
40	SET-FLOPPY-FORMAT

```

41     FORMAT-FLOPPY
42     READ-DENSITY
      READ-PARITY-AND-DENSITY
      READ-FORMAT
43     READ-DELETED-RECORD
44     WRITE-DELETED-RECORD
46     GET-CURRENT-DISK-ADDRESS

```

Further explanation of IFUNC, IPAR1 and IPAR2 and the rules for the use of each function are given under @DEVICE-FUNCTION.

**Output Parameters:**

ISTAT = function value

**Rules:**

1. Callable from background and foreground.
2. If IFUNC is in the range 5 to 24<sub>s</sub>, the parameters MARRY, MWORD and NWORD are dummy parameters.
3. If IFUNC is in the range 20 to 24<sub>s</sub>, the hardware status word is returned in the A register.

**Example:**

In Fortran:

```

      INTEGER MAGTP
      DIMENSION MARRY (N)
C      IUNIT = UNIT NO. FROM OPEN STATEMENT OR LOGICAL DEVICE NO.
      ...
      ISTAT = MAGTP(IFUNC, MARRY, IUNIT, IPAR1, IPAR2)
      IF(ISTAT.NE.0) CALL ERROR(ISTAT)

```

## MCALL (MON 132)

See also related MEXIT.

### Function:

Call a subroutine on another segment.

### Monitor call format:

	LDT	(PAR	% T = address of parameters
	MON	132	% MCALL
	...		% return here after MEXIT from subroutine
PAR,	SUBR		
	NEWSG		

### Input Parameters:

SUBR = address of subroutine.

NEWSG = new segments to be loaded as current segments in RT description.

Most significant byte = segment number 1, least significant byte = segment number 2. Byte = 377<sub>8</sub>: do not change segment.

### Output Parameters:

At entry point of subroutine (SUBR):

T = segment numbers of calling program (before new values from PAR + 1).

L = return address.

### Rules:

1. Callable only from foreground.
2. MCALL may be nested.
3. MEXIT is used for the return.
4. Reentrant segment (if any) is unchanged.

### Example:

On segment 20<sub>8</sub>:

	LDT	(PAR	% T = address of parameters
	MON	132	% MCALL
	...		% return here after MEXIT
PAR,	SUBR		% address of entry point
	10030		% new segments are 20 <sub>8</sub> and 30 <sub>8</sub>
	...		

On segment 30<sub>8</sub>:

SUBR,	STT	SAVT	% save T register
	COPY	SL DT	% save L register
	STT	SAVL	%
	...		% local processing in subroutine
	LDT	SAVL	% L = return address
	COPY	ST DL	%
	LDT	SAVT	% T = calling program segments
	MON	133	% MEXIT to location after MCALL above
SAVT,	0		
SAVL,	0		

## MDLFI (MON 54)

See also @DELETE-FILE and @DELETE-USERS-FILES.

### Function:

Delete a file and release its pages.

### Monitor call format:

MON 54

### Input Parameters:

X = address of a file name string. The string must be terminated by an apostrophe ('). The name must be unique. If a version is specified, only that version is deleted (DEF version = all versions).

### Output Parameters:

Error return: A = error number.

Skip return: OK.

### Rules:

1. Callable from background and foreground.
2. Permitted for all users with directory (D) access to the file (for foreground programs, user RT must have directory access).

### Example:

In MAC:

```

        LDX      (FNAME      % address of file name
        MON      54          % MDLFI
        JMP      ERROR       % error return
        ...
        FNAM,    ' F-1:DATA'  % file name
    All versions of the file F-1:DATA are deleted.
```

In Fortran:

```

CHARACTER FILNAM*20
DATA FILNAM/' F-1:DATA' '' /
...
CALL MDLFI(FILNAM)
IF(ERRCODE.NE.0) CALL ERROR
```

## MEXIT (MON 133)

See also MCALL.

### Function:

Exchange one or both current segments. Can be used to return to a call (by MCALL) from another segment. Execution proceeds in the logical address determined by the content of the L-register.

### Monitor call format:

LDT	MSEG	% T = new current segment no.
JPL	CHSEG	% call MEXIT routine
...		% return after monitor call (if segment not exchanged)
...		
CHSEG,	MON	133 % MEXIT
...		

### Input Parameters:

MSEG = most significant byte: first segment number; least significant byte: second segment number. Normally only one segment is exchanged at a time. Segment number 377<sub>8</sub> means do not change segment.

### Output Parameters:

None.

Return from the monitor call is made through the L-register in the calling program. I.e. if returning from a routine called by MCALL the L-register must contain the original value (point to the return address after MCALL).

### Rules:

1. Callable only from foreground.
2. Note the difference in function between MON 133 and CALL MEXIT (in Fortran).

### Example:

In MAC:

Assume current segments are 36<sub>8</sub> and 40<sub>8</sub>.

LDT	MSEG	% T = new current segment no.
JPL	CHSEG	% call MEXIT routine
...		
...		
CHSEG,	MON	133 % MEXIT
MSEG,	17440	% new first segment = 37 <sub>8</sub>

Segment 37<sub>8</sub> is loaded as new first segment.

In FORTRAN:

CALL MEXIT (37B)

Segment 37<sub>8</sub> is loaded as the new second segment.

## MGTTY (MON 16)

See also @GET-TERMINAL-TYPE. See also related MSTTY.

### Function:

Get terminal type. The terminal type is an integer value residing in the datafield of the terminal.

### Monitor call format:

MON 16

### Input Parameters:

T = logical device number (must be a terminal).

### Output Parameters:

Error return: A = error number.

Skip return: A = terminal type.

### Rules:

1. Callable from background and foreground.
2. Norsk Data's standard terminal types are shown in Appendix B.
3. If logical device number = 0 it is the user's terminal. If > 0, it has the same meaning for background and foreground programs.

### Example:

In Fortran:

```
C IUNIT - LDN OR FORTRAN I/O UNIT NO.
   CALL MGTTY(IUNIT,ITT)
   IF(ERRCODE.NE.0) CALL ERROR
```

## MHDLC (MON 201)

Special monitor call/driver for HDLC interface. Callable from background and RT-programs. See "Communication Guide", ND-60.134 for further description.



## MSG (MON 32)

See also related IOUT.

### Function:

Write a character string on the user's terminal.

### Monitor call format:

MON 32

### Input Parameters:

X = address of character string. The string is terminated by an apostrophe (').  
Maximum number of characters is 512<sub>10</sub>.

### Output Parameters:

None.

### Rules:

1. Callable only from background.
2. Parameters are fetched and returned to the user via the alternative page table.

### Example:

In MAC:

```

          LDX      (TEXT
          MON      32          % MSG
TEXT,      ' THIS IS TEXT'    % text string
```

In Fortran:

```

CHARACTER TEXT*10
DATA TEXT/'MORE TEXT'/' /
CALL MSG(TEXT)
```

## MSTTY (MON 17)

See also @SET-TERMINAL-TYPE. See also related MGTTY.

### Function:

Set terminal type. The value is placed in the terminal type location of the datafield.

### Monitor call format:

MON 17

### Input Parameters:

T = logical device number (must be a terminal).

A = terminal type ( $-32768 \leq A \leq 32767$ ).

### Output Parameters:

Error return: A = error number.

Skip return: OK.

### Rules:

1. Callable from background and foreground.
2. Normal input and output to the terminal is unaffected but the value may be used through MGTTY for terminal dependent I/O. Norsk Data's standard terminal types are shown in Appendix B.
3. See rule 3 of MGTTY (MON 16).

## MUIDI (MON 213)

### Function:

Get directory and user index.

### Monitor call format:

MON 213

### Input Parameters:

X = memory address as directory name:user name (max. 16 characters before and 16 characters after colon) or user name (max. 16 characters) string.

### Output Parameters:

Error return: A = error number.

Skip return: T = directory index (left byte) and user index (right byte).

**NOWT (MON 36)****Function:**

Set wait or no wait mode for I/O calls to character devices.

**Monitor call format:**

	LDA	(PAR	% A = address of parameter list
	MON	36	% NOWT
	STA	ISTAT	% return status
	...		
PAR,	LDN		
	IOFL		
	NOWF		

**Input Parameters:**

LDN = logical device number of a character-oriented device.  
 IOFL = I/O flag.  
       = 0: input  
       = 1: output  
 NOWF = no wait flag  
       = 0: the I/O calls cause I/O wait mode.  
       ≠ 0: the I/O calls never cause an I/O wait. Instead of setting the program in I/O wait (see rule 2 of INBT and OUTBT) the I/O call returns with error number = 3 (EOF). If the device then detects a break (restart) condition, a function similar to monitor call RT executed. If the program has become passive (RTWT or HOLD) it will be restarted immediately.

**Output Parameters:**

A = 0: OK.  
 A ≠ 0: error in parameters.

**Rules:**

Callable from background and foreground.

**N500M (MON 60)**

Special monitor call for ND-500 Monitor. Not to be used by application programs.

## OPEN (MON 50)

See also @OPEN-FILE. See also related CLOSE.

### Function:

Open a file with specified access.

### Monitor call format:

MON 50

### Input Parameters:

X  $\neq$  0: address of file name string (must be terminated by an apostrophe (')).  
     = 0: file name is read from the terminal (illegal from batch and RT).  
 A = address of default file type string (must be terminated by a single apostrophe (')).  
 T = access code.  
     = 0: sequential write  
     = 1: sequential read  
     = 2: random read or write  
     = 3: random read only  
     = 4: sequential read or write  
     = 5: sequential write append  
     = 6: random read or write common (contiguous files)  
     = 7: random read common (contiguous files)  
     = 8: random read or write (contiguous files); direct transfer mode (RFILE, WFILE and MAGTP in foreground programs)

### Output Parameters:

Error return: A = error number.

Skip return: A = logical device number (open file number).

### Rules:

Callable from background and foreground.

### Example:

In MAC:

SAT	0	% sequential write
LDX	(FILNA	% address of file name
LDA	(FILTY	% address of file type
MON	50	% open file
JMP	ERROR	% error return
...		% return when file successfully opened

FILNA, 'SCRATCH-FILE'

FILTY, 'SYMB'

## OSIZE (MON 67)

See also related ISIZE.

### Function:

Get the current number of bytes in the output buffer.

### Monitor call format:

MON 67

### Input Parameters:

T = logical device number.

### Output Parameters:

Error return: A = error number.

Skip return: A = number of bytes

### Rules:

1. Callable from background and foreground.
2. Not applicable to internal devices. Use ISIZE (MON 66) and subtract the content.

### Example:

In MAC:

SAT	3	% T = tape punch
MON	67	% OSIZE
MON	65	% error return
STA	NUMB	% normal return

In Fortran:

INTEGER OSIZE

NUMB = OSIZE(IUNIT)

## OUTBT (MON 2)

See also M8OUT and B8OUT. See also related INBT.

### Function:

Write one byte to a device or a file. If the device is a data link or word oriented internal device, write one word.

### Monitor call format:

MON 2

### Input Parameters:

T = logical device number.

A = data in least significant byte or full word if data link or word oriented internal device.

### Output Parameters:

Error return: A = error number

Skip return: no error

### Rules:

1. Callable from background and foreground.
2. If the output buffer is full, the program enters the I/O wait state. This can be changed by NOWT (MON 36).
3. When data is output to a mass storage file, the byte pointer is incremented by one.

### Example:

In MAC:

LDT	FNUM	% T register = file no.
LDA	CHAR	% A register = data
MON	2	
JMP	ERROR	% error return, A = error no.
JMP	NEXT	% normal return

In FORTRAN:

```
CALL OUTCH (IFNUM, ICHAR)
IF (ERRCODE.NE.0) STOP
```

## OUTCH

See OUTBT.

## OUTST (MON 162)

See also MSG. See also related INSTR.

### Function:

Write a string of characters to a peripheral device.

### Monitor call format:

LDA	(PAR	% A = address of parameter list
MON	162	% OUTST
STA	ISTAT	% A = function value
...		
PAR,	LDN	
	MARRY	
	MCHAR	

### Input Parameters:

LDN = logical device number of a peripheral device.

MARRY = name of data destination.

MCHAR = number of characters to be written.

### Output Parameters:

ISTAT = -1: error in parameters.  
 = 0 in bits 15, 14: OK.  
 = 2 in bits 15, 14: total size of the device buffer < MCHAR (does not apply to terminals and communication channels).  
 = 3 in bits 15, 14: device error, bit 7-0 contains error number.

### Rules:

1. Callable from background and foreground.
2. Logical device number must not be 1 (user's terminal). Use RSIO (MON 143) to get the logical device number.
3. If all characters in MARRY cannot be output to the internal output buffer, the program is set in a waiting state.
4. This is an option and must be specified when SINTRAN is ordered.
5. Parameters are fetched and returned to the user via the alternative page table.

### Example:

In Fortran:

```
INTEGER OUTST
```

```
...
```

```
ISTAT = OUTST(IUNIT, MARRY, MCHAR)
```

## PAGET (MON 57)

### Function:

Get values of the five parameters kept in the system for each background user. In the case of enabled termination handling some of the parameters are given values by SINTRAN whenever a program is terminated with escape, MON 65 or a fatal error.

### Monitor call format:

```

          LDA      (PARLI
          MON      57
          .
          .
PARLI,      ARRAY
ARRAY,      0              % p1
            0              % p2
            0              % p3
            0              % p4
            0              % p5
  
```

### Output parameters:

In the case of termination handling by SINTRAN:

p1: left byte = directory index, right byte = user index.

p2: logical device number (terminal number).

p3: -1 if escape, otherwise error number.

p4: user defined.

p5: user defined.

### Rules:

Callable only from background.

### Example: In Fortran:

```

INTEGER IARR(5)
CALL PAGET(IARR)
  
```



## PASET (MON 56)

### Function:

Set the values of the five user parameters for each background program.

### Monitor call format:

```

          LDA      (PARLI
          MON      56
          .
          .
PARLI,     ARRAY
ARRAY,     0              % p1
           0              % p2
           0              % p3
           0              % p4
           0              % p5

```

### Input Parameters:

p1: left byte = directory index, right byte = user index  
 p2: logical device number (terminal number)  
 p3: —1 if escape, otherwise error number  
 p4: user defined  
 p5: user defined

### Rules:

Callable only from background.

### Example:

```

In Fortran:
INTEGER IARR(5)
CALL  PASET(IARR)

```

## PRIOR (MON 110)

See also @PRIOR.

### Function:

Set priority of an RT program.

### Monitor call format:

LDA	(PAR	% A = address of parameter list
MON	110	% PRIOR
STA	IP	% A = the old priority
...		
PAR,	(RTPRO	
	IPRI	

### Input Parameters:

RTPRO = address of RT description. 0 = calling program.

IPRI = new priority.  $0 \leq IPRI \leq 255_{10}$ .

### Output Parameters:

IP = old priority.

### Rules:

1. Callable only from foreground.
2. Programs with priority = 0 will never be started.

### Example:

In Fortran:

IP = PRIOR (KLOKK, 30)

The priority of the RT program KLOKK is set to  $30_{10}$ .

## PRLS (MON 125)

See also @PRLS. See also related PRSRV.

### Function:

Release a device from the program which reserved it.

### Monitor call format:

	LDN	(PAR	% A = address of parameter list
	MON	125	% PRLS
	...		
PAR,	LDN		
	IOF		

### Input Parameters:

LDN = logical device number.

IOF = 0: input part.

= 1: output part.

### Output Parameters:

None.

### Rules:

Callable only from foreground.

### Example:

In Fortran:

```
CALL PRLS(410B, 1)
```

The output part of device 410B is released from the program which reserved it.

## PRSRV (MON 124)

See also @PRSRV. See also related PRLS.

### Function:

Reserve a device for an RT program.

### Monitor call format:

LDA	(PAR	% A = address of parameter list
MON	124	% PRSRV
STA	ISTAT	% A = function value
...		
PAR,	LDN	
	IOF	
	(RTPRO	

### Input Parameters:

LDN = logical device number; must be a peripheral device or a semaphore.

IOF = 0: input part.

= 1: output part.

RTPRO = address of RT description. 0 = own program.

### Output Parameters:

ISTAT = 0: reservation was successful.

< 0: device was already reserved.

### Rules:

Callable only from foreground.

### Example:

In Fortran:

EXTERNAL KLOKK

INTEGER PRSRV

...

ISTAT = PRSRV (401B, 1, KLOKK)

Output part of device 401B is reserved for the RT program KLOKK.

## QERMS (MON 65)

See also ERMSG.

### Function:

Print the error message for a given SINTRAN error number. The program is then terminated.

### Monitor call format:

MON 65

### Input Parameters:

A = error number for which message is to be printed.

### Output Parameters:

None.

### Rules:

1. Callable from background and foreground.
2. The message is output to the user's terminal, except when the program is running under a batch or mode file or foreground program, when the message is output to the error device.

### Example:

In MAC:

SAT	1	% T = user's terminal
MON	1	% INBT
MON	65	% print error message and terminate
STA	CHR	% save byte
...		

QERMS is used directly as an error return.

## RDISK (MON 5)

See also RPAGE and RFILE. See also related WDISK.

### Function:

Read random block from the scratch file, logical device number 100<sub>a</sub>.

### Monitor call format:

MON 5

### Input Parameters:

T = block number ( $T \geq 0$ )

X = memory address of data destination.

### Output Parameters:

Error return: A = error number.

Skip return: A = original contents.

### Rules:

1. Callable only from background.
2. Standard block size is 256<sub>10</sub> words. It can be changed by SETBS (MON 76).
3. RFILE can be used instead.

### Example:

In MAC:

LDT	BLNO	% T register = block no.
LDX	(DATA	% X register = memory address
MON	5	% RDISK
JMP	ERROR	% error return
...		% normal return

**REABT (MON 75)**

See also related SETBT.

**Function:**

Read the byte pointer as used by the next sequential access of mass-storage file.

**Monitor call format:**

MON 75

**Input Parameters:**

T = file number.

**Output Parameters:**

Error return: A = error number

Skip return: A = byte pointer ( $0 \leq AD \leq$  maximum byte pointer)

**Rules:**

Callable from background and foreground.

**Example:**

In MAC:

LDT	IFNO	% T = file number
MON	75	% REABT
MON	65	% error return
STD	IBYTE	% AD = byte pointer

In Fortran:

```
DOUBLE INTEGER IBYTE
OPEN(UNIT = IUNIT, ... )
CALL REABT(IUNIT, IBYTE)
IF(ERRCODE.NE.0) CALL ERROR
```

## REENT (MON 167)

### Function:

Attach a segment as a reentrant segment.

### Monitor call format:

LDA	(PAR	% A = address of parameter list
MON	167	% REENT
	...	
PAR,	ISEG	

### Input Parameters:

ISEG = segment number.

### Output Parameters:

None.

### Rules:

1. Callable from background and foreground.
2. The segment may overlap the current two segments. As long as access to a page in ISEG is only fetch and read, the page is considered reentrant. When the page is accessed for write it loses its reentrant status and is considered nonreentrant belonging to one of the two overlapping current segments. If the page is accessed for write and there is no overlapping page in the current segments, an error message is produced.
  - a) The segment is treated as reentrant only for RT-programs which have declared it reentrant.
  - b) The parts of the reentrant segment which do not overlap virtual addresses in the user's ordinary segments are read only and are intended to contain pure procedure or constants. When trying to write to these segments the error message "outside segment bounds" is given.
  - c) It is possible to write to the overlapping part. A working copy of the pages is created for each RT-program involved.

### Example:

In Fortran;  
CALL REENT (ISEG)



## RELES (MON 123)

See also @RELEASE-DEVICE-UNIT, @RELEASE-FILE and PRLS. See also related RESRV.

### Function:

Release a device so that it can be used by other programs.

### Monitor call format:

	LDN	(PAR	% A = address of parameter list
	MON	123	% RELES
	...		% normal return
PAR,	LDN		
	IOF		

### Input Parameters:

LDN = logical device number.

IOF = 0: input part.

= 1: output part.

### Output Parameters:

None.

### Rules:

1. Callable from background and foreground.
2. Normal termination of a foreground program releases the device. CLOSE (MON 43) and @CLOSE also release the device. See also rule 2 of RESRV (MON 122).

### Example:

In Fortran:

```
CALL RELES(LDN, IOF)
```

**RERRP (MON 207)****Function:**

Read-error-parameters. Returns to the user the parameters from the last system error (runtime error) plus a flag indicating whether the last terminated RT-program was aborted by the system, by itself or by other user RT-programs.

**Monitor call format:**

```

          LDA      (PARLI
          MON      207
          JAZ      OK          % A-register = 0: OK
          JMP      ERROR      % A-register ≠ 0: error
OK,
          .
          .
PARLI,    ARRAY
          .
          .
ARRAY,    0          % error number (ASCII)
          0          % error P-reg (user program)
          0          % error parameters 1 (A-reg)
          0          % error parameters 2 (T-reg)
          0          % RT-program
          0          % flag (0 if aborted by system else
                    RT-description address)

```

**Input Parameters:**

None.

**Output Parameters:**

A = 0: OK

= 153<sub>h</sub> illegal address (of parameter) reference in monitor call. See call format.

**Rules:**

1. ASCII error code does not have parity bits set.
2. ASCII error code is the decimal run-time error.
3. The first five parameters are set by the IERR routine when an RT-error occurs. The flag is only changed if RT-termination handling is enabled. In such cases it is set to zero if the program is aborted by the system or set to RT-description address if terminated by MON 105 (ABORT) or MON 0.

## RESRV (MON 122)

See also @RESERVE-DEVICE-UNIT, @RESERVE-FILE and PRSRV. See also related RELES.

### Function:

Reserve a device for the exclusive use of the calling RT program.

### Monitor call format:

	LDA	(PAR	% A = address of parameter list
	MON	122	% RESRV
	STA	ISTAT	% A = reservation status
	...		
PAR,	LDN		
	IOF		
	IRET		

### Input Parameters:

LDN = logical device number; must be a peripheral device or a semaphore.

IOF = 0: input part.

= 1: output part.

IRET = 0: if the unit is reserved, set the program in a waiting state. Otherwise, return from the call.

= 1: if the unit is reserved, return with ISTAT < 0; otherwise return with ISTAT = 0.

### Output Parameters:

ISTAT = output status, dependent on IRET as shown above.

### Rules:

1. Callable from background and foreground.  
When used in a background program the unit is still reserved even if the user program is terminated. The terminal user can release the unit by @PRLS under user SYSTEM; it is also released at log out.  
When used in an RT program the unit is released when the program is terminated.
2. OPEN (MON 50) and @OPEN reserve a device. If OPEN is used from background the device is released if escape is pressed; if RESRV is used it is not released. Internally, OPEN uses the file system, RESRV does not.

## RFILE (MON 117)

See also @RFILE and RPAGE. See also related WFILE.

### Function:

Transfer a number of words from one or more random file blocks into memory.

### Monitor call format:

	LDA	(PAR	% address of parameter list
	MON	117	% RFILE
	JAF	ERROR	% error if A $\neq$ 0, A = error no.
	...		% OK if A = 0
PAR,	IFNO		
	IRET		
	MADR		
	IBNO		
	NWORD		

### Input Parameters:

IFNO = file number from OPEN monitor call.

IRET = 0: return from RFILE is suspended until the transfer is complete.  
 $\neq$  0: return is made as soon as the transfer is started. WAITF (MON 121) can be used to check for completion. IRET is always 0 for background programs.

MADR = memory address of data destination.

IBNO  $\geq$  0: number of file block where data starts.  
 = -1: read next block.

NWORD = number of words to be transferred. If NWORD is greater than the block size, the next block or blocks are read. In direct transfer mode (see OPEN) NWORD must be a multiple of the block size as set by SETBS (MON 76).

### Output Parameters:

Error return: A = error number.  
 Skip return: OK.

### Rules:

1. Callable from background and foreground.
2. This monitor call can be used as a substitute for RDISK and RPAGE.
3. Parameters are fetched and returned to the user via the alternative page table.
4. When using access code D (direct transfer) the logical block size may be a multiple of the physical block size.
5. On a peripheral file the next block is always read, ie. reading is sequential.
6. Data transfer across segment or RT-common limits is not allowed.

## RMAX (MON 62)

See also @FILE-STATISTICS. See also related SMAX.

### Function:

Get the number of data bytes in an open file.

### Monitor call format:

MON 62

### Input Parameters:

T = file number.

### Output Parameters:

Error return: A = error number.

Skip return: AD = number of bytes.

### Rules:

1. Callable from background and foreground.
2. The number of data bytes is only relevant for sequentially accessed files.

### Example:

In MAC:

LDT	(101	% T = file number
MON	62	% RMAX
JMP	ERROR	% error return
...		% normal return

In Fortran:

```
DOUBLE INTEGER NOBYT
OPEN(UNIT = IUNIT, ...)
CALL RMAX(IUNIT, NOBYT)
IF(ERRCODE.NE.0) CALL ERROR
```

## ROBJE (MON 41)

See also @DUMP-OBJECT-ENTRY.

### Function:

Read object entry of an opened file.

### Monitor call format:

MON 41

### Input Parameters:

T = opened file number.

A = memory address of data destination.

### Output Parameters:

Error return: A = error number.

Skip return: OK, data destination contains object entry.

### Rules:

1. Callable from background and foreground.
2. 40<sub>s</sub> words are read.
3. The device number location of the object entry contains the logical device number (bits 11-0) and the unit number (bits 15-12) of the mass storage device on which the file resides.

### Example:

In MAC:

(file 101 is opened)

LDT	(101	% open file no. = 101
LDA	(OBJEN	% read to data area
MON	41	% ROBJE
JMP	ERROR	% error return
...		% normal return

## RPAGE (MON 7)

See also RDISK and RFILE. See also related WPAGE.

### Function:

Read random block from a file.

### Monitor call format:

MON 7

### Input Parameters:

T = file number ( $100_8 \leq T \leq 121_8$ )

A = block number ( $A \geq 0$ )

X = memory address of data destination.

### Output Parameters:

Error return: A = error number.

Skip return: A = original contents.

### Rules:

1. Callable from background and foreground.
2. Standard block size is  $256_{10}$  words. It can be changed by SETBS (MON 76).
3. This monitor call can be substituted by RFILE.

### Example:

In MAC:

LDT	FILE	% T register = file no.
LDA	BLNO	% A register = block no.
LDX	(DATA	% X register = memory address
MON	7	%RPAGE
JMP	ERROR	% error return
...		% normal return

## RSEGM (MON 53)

See also @LIST-SEGMENT and @SINTRAN-SERVICE-PROGRAM,  
\*DUMP-SEGMENT-TABLE-ENTRY.

### Function:

Read the contents of a segment table entry to the user's memory (option).

### Monitor call format:

LDA	(PAR	% A = address of parameter list
MON	53	% RSEGM
MON	65	% error return
...		% normal return
PAR,	SEGNO	
	ARRAY	
	...	

### Input Parameters:

SEGNO = segment number; if = 0 means RT-COMMON in which case only  
LOGADR, MADR (first physical page) and FLAG are significant.

### Output parameters:

ARRAY = 5 word array containing  
 SEGLINK  
 BPAGLINK  
 LOG ADR  
 MADR  
 FLAG

### Rules:

Callable from background and foreground.



## RSIO (MON 143)

**Function:**

Get execution mode of calling background program, user and input and out file numbers.

**Monitor call format:**

MON 143

**Input Parameters:**

None.

**Output Parameters:**

A = execution mode

= 0: interactive program

= 1: batch job

= 2: @MODE job

T = file number of command input file (logical unit number of terminal in interactive mode).

D = file number of command output file (logical unit number of terminal in interactive mode).

X = main directory index (most significant byte) and user number (least significant byte) under which the program is running (corresponds to the number listed by @LIST-USERS)

**Rules:**

Callable only from background.

**RSPQE (MON 55)****Function:**

Read next spooling queue entry. The entry is removed from the queue.

**Monitor call format:**

MON 55

**Input Parameters:**

X = memory address of queue entry destination. This should be a 128<sub>10</sub> word area.

T = logical device number of spooling device.

**Output Parameters:**

Error return: A = error number.

Skip return: OK.

**Subroutine call format:**

Not available.

**Rules:**

1. Callable from background and foreground.
2. The layout of the queue entry is (decimal values):
  - Word 1; number of printed copies
  - Word 2; print text flag. If ASCII apostrophe, then text will be printed independent of spooling conditions.
  - Words 3-49; file name of spooling file.
  - Words 50-128; user text to be printed on the error device.

**Example:**

In MAC:

LDX	(SQE	% X = address of spooling queue entry
LDT	(5	% T = logical device number
MON	55	% RSPQE
JMP	ERROR	% error return
...		% normal return
SQE,	*+200/	

## RT (MON 100)

See also @RT. See also related SET and ABSET.

### Function:

Put an RT program in the execution queue.

### Monitor call format:

LDA	(PAR	% A = address of parameter list
MON	100	% RT
...		
PAR,	(RTPROG	

### Input Parameters:

RTPROG = address of RT description. 0 = calling program.

### Output Parameters:

None.

### Rules:

1. Callable only from foreground.
2. If the program is already in the execution queue, it is set for repeated execution.

### Example:

In Fortran:  
 EXTERNAL KLOKK  
 ...  
 CALL RT(KLOKK)

## RTDSC (MON 27)

See also related GETRT and @LIST-RT-DESCRIPTION.

### Function:

Read the RT description of a program.

### Monitor call format:

LDA	(PAR	% A = address of parameter list
MON	27	% RTDSC
MON	65	% error return
STA	NDEV	% A = no. of connected devices
...		
PAR,	(RTPRO	
	IARR	

### Input Parameters:

RTPRO = address of RT description. 0 = calling program.

IARR = array where RT description is to be stored; size is 26<sub>10</sub> words.

### Output Parameters:

Error return: A = error number.

Normal return: A = number of devices connected to the RT program by monitor call CONCT.

### Rules:

Callable only from foreground.

### Example:

In Fortran:

INTEGER RTDSC

EXTERNAL RTPRO

DIMENSION IARR(26)

NDEV = RTDSC(RTPRO, IARR)

## RTEXT (MON 134)

See also LEAVE. See also related ABORT.

### Function:

Terminate the calling RT program and release its resources.

### Monitor call format:

MON	134	% RTEXT
...		% no return

### Input Parameters:

None.

### Output Parameters:

None.

### Rules:

1. Callable from background and foreground.
2. In batch jobs this call aborts the batch job; in other cases it has the same function as LEAVE.

## RTOFF (MON 137)

See also @RTOFF. See also related RTON.

### Function:

Disable the starting of an RT program until RTON is called.

### Monitor call format:

LDA	(PAR	% A = address of parameter list
MON	137	% RTOFF
...		% normal return
PAR,	(RTPRO	

### Input Parameters:

RTPRO = address of RT description. 0 = calling program.

### Output Parameters:

None.

### Rules:

Callable only from foreground.

## RTON (MON 136)

See also @RTON. See also related RTOFF.

### Function:

Enable the starting of an RT program after a call to RTOFF.

### Monitor call format:

LDA	(PAR	% A = address of parameter list
MON	136	% RTON
...		% normal return
PAR,	(RTPRO	

### Input Parameters:

RTPRO = address of RT description.

### Output Parameters:

None.

### Rules:

Callable only from foreground.

## RTWT (MON 135)

See also related HOLD.

### Function:

Set the RT program in a state where it waits to be restarted. The next time it is started (eg. by calling RT from another program) it continues after the call to RTWT.

### Monitor call format:

MON	135	% RTWT
...		% continue at this address

### Input Parameters:

None.

### Output Parameters:

None.

### Rules:

1. Callable only from foreground.
2. No resources are released.
3. RT-description repeat bit overrides HOLD.

## RUSER (MON 44)

See also @DUMP-USER-ENTRY.

### Function:

Read a user entry from a directory.

### Monitor call format:

MON 44

### Input Parameters:

A = memory address of data destination.

X = address of character string containing user name terminated by an apostrophe (').

### Output Parameters:

Error return: A = error number.

Skip return: A = OK.

### Rules:

1. Callable from background and foreground. User SYSTEM may read any user entry. Other users may read their own entry. Foreground programs may read user RT entry.
2. 40<sub>8</sub> words are read.

### Example:

In MAC:

	LDA	(USEN	% read to data area
	LDX	(NAME	% user name
	MON	44	% RUSER
	JMP	ERROR	% error return
	...		% normal return
USEN,	*+40/		% data
NAME,	' GUEST'		% user name

**SBRK (MON 47)****Function:**

Breakpoint monitor call. Used only by MAC assembler.

**SET (MON 101)**

See also @SET. See also related ABSET.

**Function:**

Enter an RT program in the time queue and transfer it to the execution queue after a specified period.

**Monitor call format:**

	LDA	(PAR	% A = address of parameter list
	MON	101	% SET
		...	
PAR,	(RTPRO		
	NOTIM		
	ITUN		

**Input Parameters:**

RTPRO = address of RT-description. 0 = calling program.

NOTIM = the number of time units the program is to stay in the time queue.

ITUN = 1: basic time units (normally 20 msec)

= 2: seconds

= 3: minutes

= 4: hours

**Output Parameters:**

None.

**Rules:**

1. Callable only from foreground.
2. If the program is already in the time queue it is removed and reinserted according to the new specifications.
3. Programs in the time queue scheduled by SET are not affected when the clock is adjusted (CLADJ).
4. If  $\text{NOTIM} \leq 0$  the program is transferred from the time queue to the execution queue the next time the basic time unit counter is incremented.

**Example:**

In Fortran:

```
EXTERNAL KLOKK
```

```
...
```

```
CALL SET(KLOKK, 5, 3)
```

The RT program KLOKK is set up for execution in 5 minutes.



## SETBL (MON 77)

See also @SET-BLOCK-POINTER.

### Function:

Set the byte pointer to the beginning of a block.

### Monitor call format:

MON 77

### Input Parameters:

T = file number.

A = block number (  $A \leq 0$  ).

### Output Parameters:

Error return: A = error number.

Skip return: A = original contents.

### Rules:

1. Callable from background and foreground.
2. The new byte pointer is the product of block number and size.

### Example:

In MAC:

LDT	IFNO	% T = file no.
LDA	IBNO	% A = block no.
MON	77	% SETBL
MON	65	% error return
...		% OK return

## SETBS (MON 76)

See also @SET-BLOCK-SIZE. See also related SETBL, RFILE, WFILE.

### Function:

Set (temporarily) the block size of a file. The value is used for random read and write.

### Monitor call format:

MON 76

### Input Parameters:

T = file number.

A = block size in words.

### Output Parameters:

Error return: A = error number.

Skip return: A = original contents.

### Rules:

1. Callable from background and foreground.
2. Files are allocated in pages of 1K words. A block size > 1K words forces the operating system to read or write several pages for the monitor calls RFILE and WFILE. Factors of 1024 are more efficient than other sizes.
3. The new size is only valid as long as the file is open. When a file is opened again the block size is by default 256<sub>10</sub> words.

### Example:

In MAC:

LDT	IFNO	% T = file number
LDA	IBSIZ	% A = new block size
MON	76	% SETBS
MON	65	% error return
...		% OK return

## SETBT (MON 74)

See also @SET-BYTE-POINTER. See also related REABT.

### Function:

Set byte pointer to be used by the sequential accessing of a mass-storage file.

### Monitor call format:

MON 74

### Input Parameters:

T = file number.

AD = byte pointer ( $0 \leq AD \leq \text{maximum byte pointer}$ ).

### Output Parameters:

Error return: A = error number.

Skip return: AD = not changed.

### Rules:

Callable from background and foreground.

### Example:

In MAC:

LDT	IFNO	% T = file number
LDD	IBYTE	% AD = byte pointer
MON	74	% SETBT
MON	65	% error return
...		% normal return

In Fortran:

```
DOUBLE INTEGER IBYTE
OPEN(UNIT = IUNIT,...)
CALL SETBT(IUNIT,IBYTE)
IF(ERRCODE.NE.0) CALL ERROR
```

## SETCM (MON 12)

See also COMND.

**Function:**

Transfer string to the command input buffer.

**Monitor call format:**

MON 12

**Input Parameters:**

A = memory address of string. The string must be terminated by a single apostrophe.

**Output Parameters:**

None.

**Rules:**

1. Callable only from background.
2. Parameters are fetched and returned to the user via the alternative page table.

## SMAX (MON 73)

See also related RMAX.

### Function:

Set the value of the maximum byte pointer of a file.

### Monitor call format:

MON 73

### Input Parameters:

T = file number.

AD = maximum byte pointer.

### Output Parameters:

Error return: A = error number

Skip return: A = original contents

### Rules:

1. Callable from background and foreground.
2. Maximum byte pointer is only used for sequential access.
3. MAXBY is related to NOBYT in the call RMAX;  
MAXBY = NOBYT-1.
4. Attempts to read beyond the maximum byte pointer give an error return with error code 3 (EOF).
5. File must be opened for write.

### Example:

In MAC:

LDT	IFNO	% T = file no.
LDD	MAXBY	% AD = max. byte pointer
MON	73	% SMAX
MON	65	% error return
...		% normal return

In Fortran:

```
DOUBLE INTEGER MAXBY
OPEN(UNIT = IUNIT, ...)
CALL SMAX(IUNIT, MAXBY)
IF(ERRCODE.NE.0) CALL ERROR
```

## SPCLO (MON 40)

See also CLOSE.

### Function:

Close a spooling file. A text string may be appended to be output to the error device when the file is to be printed.

### Monitor call format:

MON 40

### Input Parameters:

T = file number ( $100_a \leq T \leq 121_a$ ).

A = 0: text is written only if required by

@DEFINE-SPOOLING-CONDITIONS.

A  $\neq$  0: text is printed unconditionally. Stop print condition occurs before printing is started.

D = number of print copies.

X = address of text string.

### Output Parameters:

Error return: A = error number.

Skip return: OK.

### Rules:

1. Callable from background and foreground.
2. If the file is not a spooling file, a normal file close occurs.

### Example:

In MAC:

LDT	(115	% close file 115 <sub>a</sub>
SAA	0	% print text only if set by @S-S-C
SAD	1	% one print copy
LDX	(TEXT	% text to be written
MON	40	% SPCLO
JMP	ERROR	% error return
...		% OK

## **SREEN (MON 212)**

**Function:**

This has the same function as REENT (MON 167) except that "shadow" pages with the "written in page" bit set will be written to mass-storage.

**Monitor call format:**

MON 212

**Input Parameters:**

None.

**Output Parameters:**

None.

**Rules:**

This is an option available in SINTRAN.

## TERMO (MON 52)

See also @TERMINAL-MODE.

### Function:

Set communication mode for any terminal.

### Monitor call format:

LDA	(PAR	
MON	52	% TERMO
JAF	ERROR	% A $\neq$ 0: A = error no.
...		% A = 0: OK
PAR,	LDN	% address of logical device no.
	MODE	% communication mode, see rule 2

### Input Parameters:

LDN = logical device number (terminal number, 0 = user's terminal).

MODE = see below.

### Output Parameters:

A = 0: OK

= -1: "USE LOCAL MODE"

$\neq$  0 and -1: bits 14-0 contain file system error code.

### Rules:

1. Callable from background and foreground.
2. The meaning of MODE is:

Mode	Capital letters?	Delay after CR?	Stop on full page?	Logout on missing carrier?
0	N	N	N	N
1	Y	N	N	N
2	N	Y	N	N
3	Y	Y	N	N
4	N	N	Y	N
5	Y	N	Y	N
6	N	Y	Y	N
7	Y	Y	Y	N
8	N	N	N	Y
9	Y	N	N	Y
10	N	Y	N	Y
11	Y	Y	N	Y
12	N	N	Y	Y
13	Y	N	Y	Y
14	N	Y	Y	Y
15	Y	Y	Y	Y

3. If connected to a remote computer an error code of -1 is returned. The monitor call is only effective if executed on the local computer.

### Example:

In Fortran:

CALL TERMO(0,4)

Set "stop on full page" for user's terminal (in background). All other parameters are set to N.



## TIME (MON 11)

See also CLOCK and @DATCL. See also related UPDAT and CLADJ.

**Function:**

Read current internal time.

**Monitor call format:**

MON 11

**Input Parameters:**

None.

**Output Parameters:**

AD = time in basic time units (normally 20 msec).

**Rules:**

1. Callable from background and foreground.
2. The internal time is set to zero each time SINTRAN is started.

**Example:**

In Fortran:

```
DOUBLE INTEGER TIME, TI
TI = TIME(DUMMY)
```

## TRACB (MON 156)

**Function:**

Optional general monitor call for NORCOM tracker ball (phased-out product). TRACB documentation is given in previous versions of "SINTRAN III User's Guide".

## TUSED (MON 114)

See also @TIME-USED.

**Function:**

Read the CPU time used by the terminal since last log on or start of batch job.

**Monitor call format:**

MON 114

**Input Parameters:**

None.

**Output Parameters:**

AD = CPU time as double word.

**Rules:**

Callable only from background.

**Example:**In Fortran:

```
DOUBLE INTEGER TD, TUSED
TD = TUSED(DUMMY)
```

## UNFIX (MON 116)

See also @UNFIX. See also related FIX and FIXC.

**Function:**

Permit a previously fixed segment to be swapped out.

**Monitor call format:**

LDA	(PAR	% A = address of parameter list
MON	116	% UNFIX
...		% normal return
PAR,	ISEG	

**Input Parameters:**

ISEG = segment number.

**Output Parameters:**

None.

**Rules:**

Callable only from foreground.

## UPDAT (MON 111)

See also @UPDAT. See also related CLADJ, CLOCK and TIME.

### Function:

Give new values to the clock and calendar.

### Monitor call format:

	LDA	(PAR	% A = address of parameter list
	MON	111	% UPDAT
	...		% normal return
PAR,	MIN		
	Ihour		
	IDAY		
	MONTH		
	IYEAR		

### Input Parameters:

MIN, Ihour, IDAY, MONTH, IYEAR = new values of clock and calendar (IYEAR must be 4 digits).

### Rules:

1. Callable only from foreground.
2. If a parameter is specified outside its range (e.g., minute > 60), an error message is output and the calling program is aborted.
3. On NORD-100, the panel clock is updated if installed.

### Example:

In Fortran:

```
CALL UPDAT(MIN, Ihour, IDAY, MONTH, IYEAR)
```

```
MIN = 6
```

```
Ihour = 11
```

```
IDAY = 13
```

```
MONTH = 3
```

```
IYEAR = 1981
```

Time and calendar is set to March 13, 1981 at 11:06.

**US0 to US7 (MON 170 to MON 177)****Function:**

Numbers reserved for user defined monitor calls.

**Monitor call format:**

MON 17n % n is 0, 1, 2, ... or 7

**Input Parameters:**

User defined.

**Output Parameters:**

User defined.

**Rules:**

1. Callable from background and foreground.
2. User defined monitor calls may be included by  
\*DEFINE-USER-MONITOR-CALL in @SINTRAN-SERVICE-PROGRAM.

## WAITF (MON 121)

### Function:

Check that mass-storage transfer is complete in RFILE or WFILE.

### Monitor call format:

	LDA	(PAR	% A = address of parameter list
	MON	121	% WAITF
	STA	ISTAT	% A = function value
	...		
PAR,	IFNO		
	IRET		

### Input Parameters:

IFNO = file number from OPEN command or monitor call.

IRET = 0: return is made from the call only if the transfer is complete; if not, the RT program is set in the I/O wait state.

≠ 0: immediate return with the function value showing the state of transfer.

### Output Parameters:

ISTAT = -1: transfer is unfinished (only when IRET ≠ 0).

= 0: transfer is finished.

> 0: error code (from WAITF, not RFILE/WFILE).

### Rules:

Callable only from foreground.

### Example:

In Fortran:

```
INTEGER WAITF
```

```
...
```

```
ISTAT = WAITF(FNUM, 0)
```

If the transfer is not finished the calling program is set in a waiting state.

## WDISK (MON 6)

See also WPAGE and WFILE. See also related RDISK.

### Function:

Write random block to the scratch file, logical device number 100<sub>8</sub>.

### Monitor call format:

MON 6

### Input Parameters

T = block number ( $T \geq 0$ ).

X = memory address of data source.

### Output Parameters:

Error return: A = error number.

Skip return: A = original contents.

### Rules:

1. Callable only from background.
2. Standard block size is 256<sub>10</sub> words. It can be changed by SETBS (MON 76).
3. WFILE can be used instead.

### Example:

In MAC:

LDT	BLNO	% T register = block no.
LDX	(DATA	% X register = memory address
MON	6	% WDISK
JMP	ERROR	% error return
...		% normal return

## WFILE (MON 120)

See also @WFILE and WPAGE. See also related RFILE.

### Function:

Write a number of words to one or more random blocks in a file.

### Monitor call format:

LDA	(PAR	% address of parameter list
MON	120	% WFILE
JAF	ERROR	% error if A $\neq$ 0, A = error no.
...		% OK if A = 0
PAR,	IFNO	
	IRET	
	MADR	
	IBNO	
	NWORD	

### Input Parameters:

IFNO = file number from OPEN command or monitor call.

IRET = 0: return from WFILE is suspended until the transfer is completed.

$\neq$  0: return is made as soon as the transfer is started. WAITF (MON 121) can be used to check for completion. IRET is always 0 for background programs.

MADR = memory address of data.

IBNO = -1: write next record.

$\geq$  0: block number where writing started.

NWORD = number of 16 bit words in the extent. Must be a multiple of the block size for direct transfer mode.

### Output Parameters:

A = 0: OK.

A  $\neq$  0: error number.

### Rules:

1. Callable from background and foreground.
2. This monitor call can be used instead of WDISK and WPAGE.
3. Parameters are fetched and returned to the user via the alternative page table.
4. When using access code D (direct transfer) the logical block size may be a multiple of the physical block size.

## WHDEV (MON 140)

See also @LIST-DEVICE.

### Function:

Check whether a logical device is reserved.

### Monitor call format:

LDA	(PAR	% A = address of parameter list
MON	140	% WHDEV
STA	I PROG	% A = address of RT description
	...	
PAR,	LDN	
	IOPA	

### Input Parameters:

LDN = logical device number.

IOPA = 0: input part.

= 1: output part.

### Output Parameters:

I PROG  $\neq$  0: address of RT description of the program which reserved the logical device.

= 0: logical device is free.

### Rules:

Callable only from foreground.



## WPAGE (MON 10)

See also WDISK and WFILE. See also related RPAGE.

### Function:

Write random block to a file.

### Monitor call format:

MON 10

### Input Parameters:

T = file number ( $100_8 \leq T \leq 121_8$ ).

A = block number ( $A \geq 0$ ).

X = memory address of data destination.

### Output Parameters:

Error return: A = error number.

Skip return: A = original contents.

### Rules:

1. Callable from background and foreground.
2. Standard block size is  $256_{10}$  words. It can be changed by SETBS (MON 76).
3. This monitor call can be substituted by WFILE.

### Example:

In MAC:

LDT	FILE	% T register = file no.
LDA	BLNO	% A register = block no.
LDX	(DATA	% X register = memory address
MON	10	% WPAGE
JMP	ERROR	% error return
...		% normal return

**WRQI (MON 163)****Function:**

Place the calling program in a waiting state until a request for input message is received from the remote computer. Used by NORDNET communication programs. If the program is interactive it may be useful not to start terminal input echoing before the receiving program asks for input.

**Monitor call format:**

MON 163

**Input Parameters:**

T = channel number.

**Output Parameters:**

Error return: A = error number.

Skip return: OK.

**Rules:**

Callable from background and foreground.

**WSEG (MON 164)****Function:**

Write all pages with the written-in-page-bit set to a segment on mass-storage.

**Monitor call format:**

LDA	(PAR	% A = address of parameter list
MON	164	% WSEG
	...	
PAR,	ISEG	

**Input Parameters:**

ISEG = segment number.

**Output Parameters:**

None

**Rules:**

Callable only from foreground.

**XMSG (MON 200)**

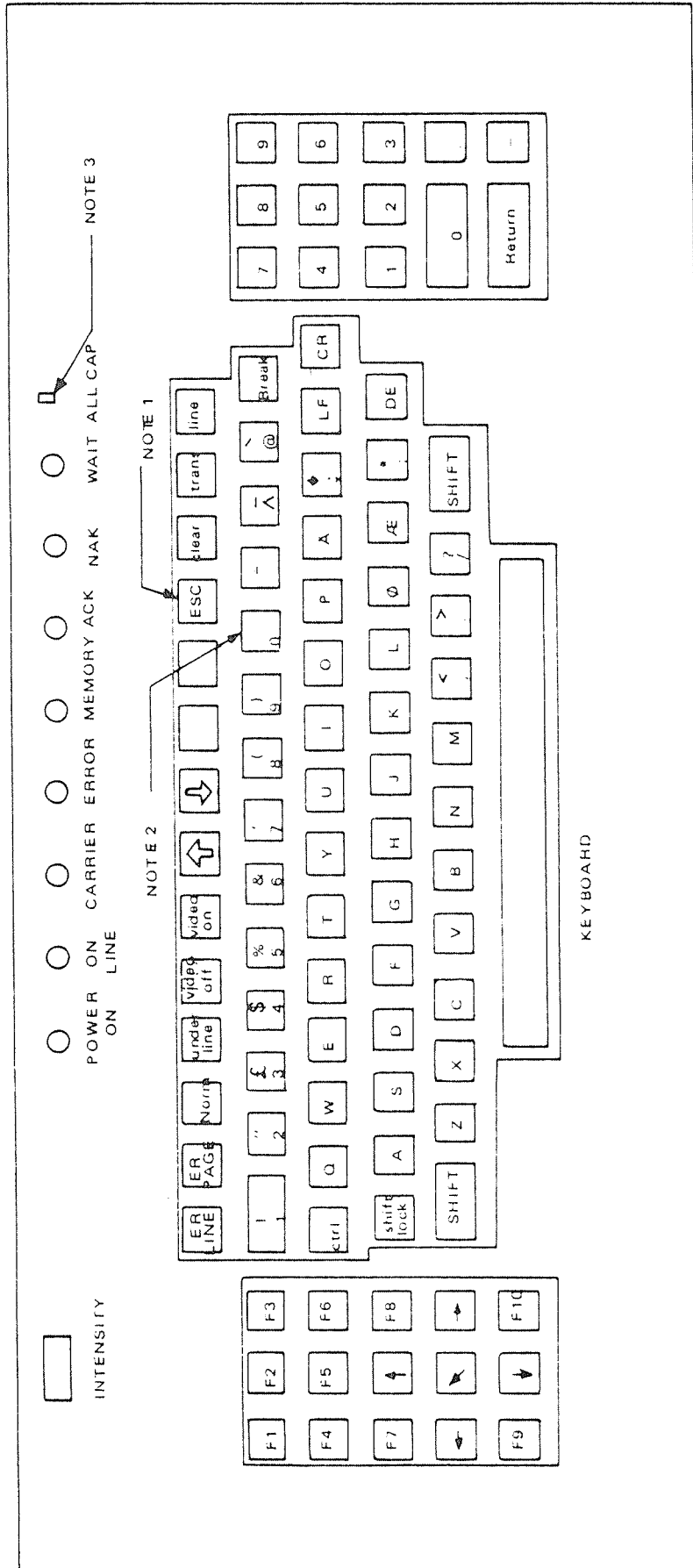
X-message. Interface to process communication system. Callable from background, RT-programs, direct-task and drivers. See "Communication Guide", ND-60.134.01 for further description.

## APPENDIX A

### KEYBOARD LAYOUT OF STANDARD TERMINALS

The following pages show the keyboard layout of the most common terminals. Some keyboards have particular properties which are documented in notes below each figure.

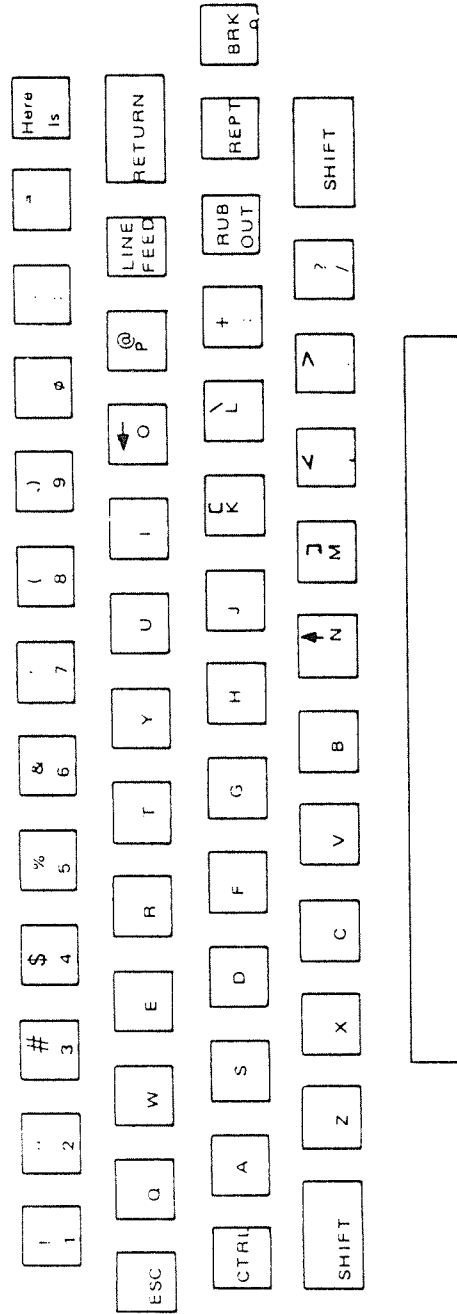
## ND No. 224N — Display Terminal Tandberg TDV 2115 (Veritas MK II)



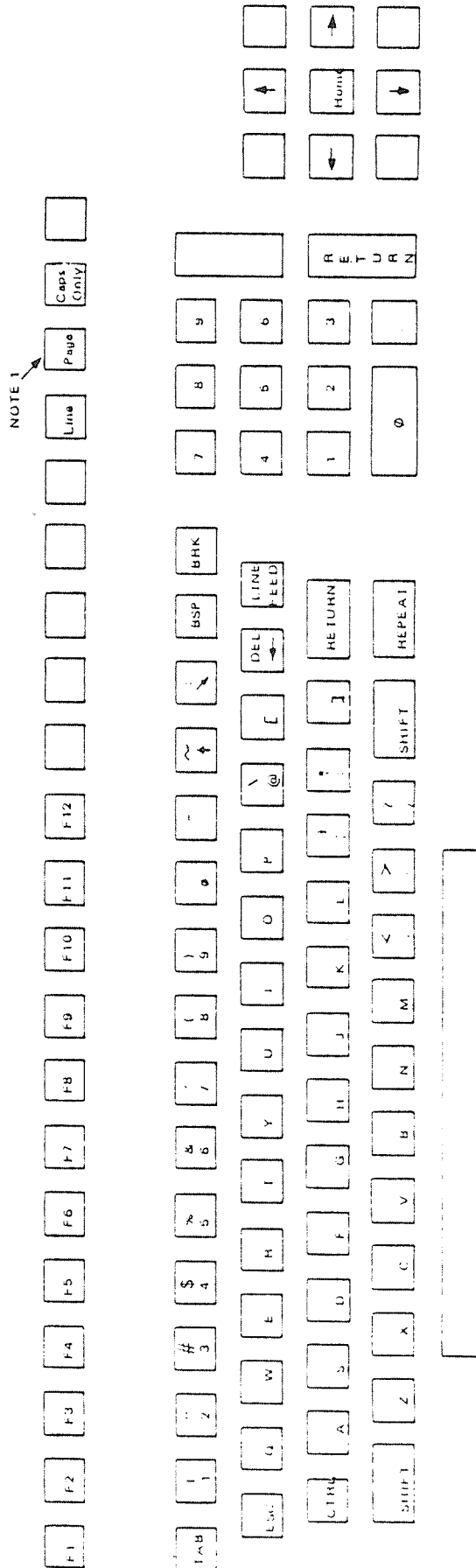
## Notes:

1. The ESC key may not have a label.
2. The underline, SHIFT/0 key is displayed as a filled square.
3. When the ALL CAP switch is ON all characters are entered as upper case. It can be moved with a pencil point.

ND No. 226E — Display Terminal INFOTON 200/1



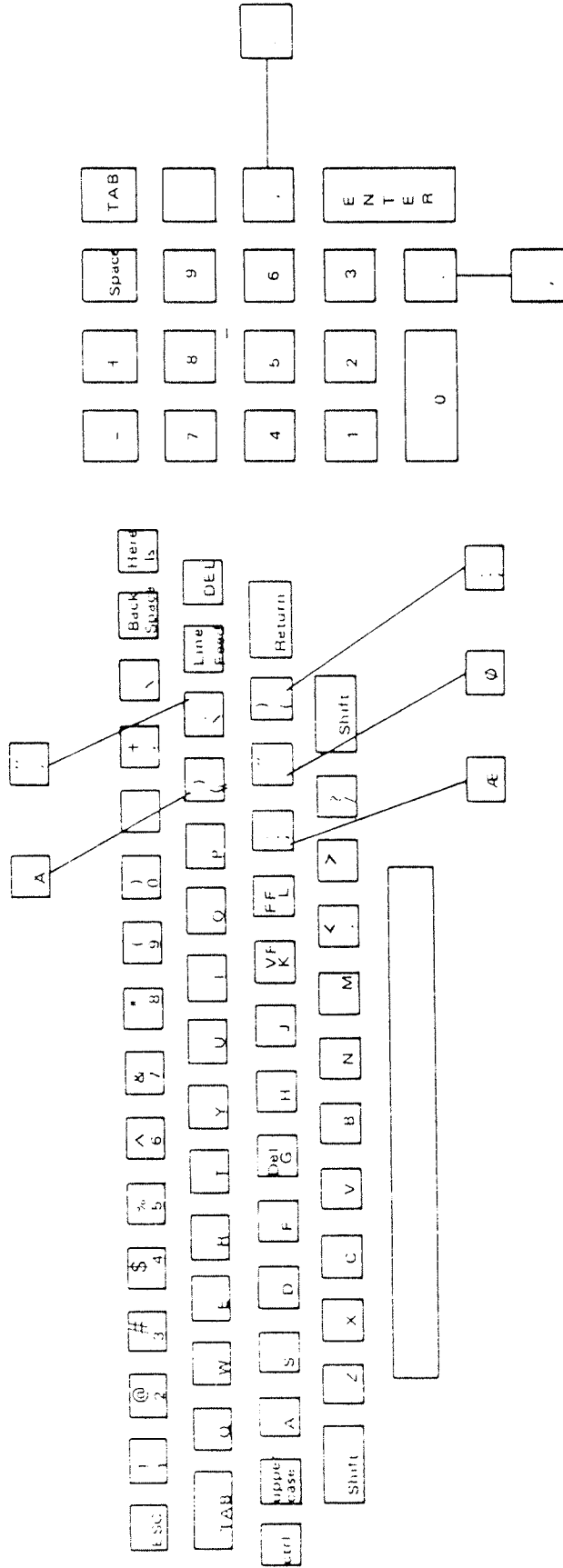
ND No. 227X -- Display Terminal INFOTON 200/4



Notes:

1. Always leave page ON.

ND No. 230 ... Texas Instrument KSR 825 with Denmark/Norway Character Set  
Option



Keytop	T1 Part No.
A	22 10282 0002
..	(Keyboard)
Æ	22 10282 0003
Ø	22 10282 0008
.	(Keyboard)

## APPENDIX B

### ND TERMINAL TYPES

The list shows the current types defined by ND. New numbers are allocated as ND adds new terminal types to systems.

The terminal type number is calculated as follows:

Bit

- |     |  |
|-----|--|
| 15  | Reserved.  |
| 14  | Set to one if the terminal is a VDU (not hard copy).                                       |
| 13  | Set to one if the terminal handles the ASCII backspace character (BS) properly.            |
| 12  | Set to one if ASCII form feed (FF) gives new page or clear screen.                         |
| 11  | Set to one if the VDU has cursor positioning (either directly or by use of cursor arrows). |
| 10  | Set if the terminal utilizes ASCII escape (ESC) within input sequences.                    |
| 7-0 | Terminal model number. See next page.  |

**Example:**

The number for TANDBERG-TDV2115-STD on logical device number 49 is set as follows:

```
@SET-TERMINAL-TYPE 49,164003B
```

or more permanently

```
@SINTRAN-SERVOCE-PROGRAM
```

```
*CHANGE-DATAFIELD <LDN> INPUT YES YES YES
```

```
CTTYP/164003
```

```
EXIT
```



## ND Terminal Types

Model Name	V	B	F	Model No.	ND No.	Comments
DUMMY				0		Terminal type not set
VISTAR-OLD	x		x	1		
TELETYPE-ASR-33				2	202	
TANDBERG-TDV2115-STD	x	x		3	224N	
INFOTON-200-1	x	x	x	4	226E	
INFOTON-400-4	x	x	x	5	228X	
DEC-VT100	x	x		6		80-column mode
TANDBERG-TDV2000	x	x		7	208	
BEEHIVE-100	x			8		
ND-NCT	x	x	x	9	680	
HAZELTINE-1520	x	x		10		
DEC-LA36		x		11	206X	Decwriter-II
VISTAR-GTX	x		x	12	210E	
ND-TET				13	212N	
TANDBERG-TDV2000-3270				14	213	
TANDBERG-TDV2115-SDS77				15	220N	
TANDBERG-TDV2115-SDS78				16	221X	
TANDBERG-TDV2115-3270				17	225X	
INFOTON-200-4				18	227X	
TEKTRONIX-4006				19	214	
TEKTRONIX-4012				20	215	
TEKTRONIX-4014-1				21	216	
PERKIN-ELMER-BANTAM-550				22		
PERKIN-ELMER-OWL-1200				23		
HEWLETT-PACKARD-2640-B				24		
HEWLETT-PACKARD-2641/2645-B				25		
BEEHIVE-150				26		
BEEHIVE-MICROBEE-1A				27		
BEEHIVE-MICROBEE-2				28		
DEC-VT52	x	x		29		
TEC-501				30		
DACOLL-242				31		
NEWBURY-7000				32		
TELEVIDEO-912/920				33		
VISUAL-200	x	x		34		
LEAR-SIEGLER-ADM-3A	x	x		35		
TDV2215-EXTENDED				36	44B	
VOLKER-CRAIG-VC404	x	x		37		
VOLKER-CRAIG-VC410	x	x	x	38		
VOLKER-CRAIG-VC414	x	x		39		
HEWLETT-PACKARD-2621A	x	x		40		
DATA-MEDIA-ELITE-3045	x	x		41		
BEEHIVE-MINIBEE	x			42		
PERICOM-6800				43		80-column mode
LEAR-SIEGLER-ADM-31				44		
BEEHIVE-DM5A				45		
FACIT-4420				46		
ADDS-VIEWPOINT				47		
HAZELTINE-EXECUTIVE-80				48		
AMPEX-DIALOGUE-80				49		
VOLKER-CRAIG-VC4404				50		

## APPENDIX C

## LOGICAL DEVICE NUMBERS USED IN SINTRAN III

<i>Octal Logical Device No.:</i>	<i>Decimal Logical Device No.:</i>	<i>Device Name:</i>
0-77	0-63	Character devices
100-177	64-127	Mass-storage files
200-277	128-191	Internal devices
300-377	192-255	Semaphores
400-477	256-319	Process Control Devices/Connect Devices
500-577	320-383	System devices
600-677	384-447	SINTRAN III/SINTRAN III Communication Devices
700-777		NORDCOM devices
1000-1077		Extension of character devices
1100-1177		System devices
1200-1277		System devices
1300-1377		System devices

<i>Octal Logical Device No.:</i>	<i>Decimal Logical Device No.:</i>	<i>Device Name:</i>
0	0	INBT;INCH (background): edited input, else: dummy
1	1	Background: own terminal RT: Terminal 1
2	2	Tape reader 1
3	3	Tape punch 1
4	4	Card reader 1
5	5	Line printer 1
6	6	Synchronous modem 1
7	7	Terminal 17
10	8	Plotter 1
11	9	Terminal 2
12	10	Tape reader 2
13	11	Tape punch 2
14	12	Bus switch device
15	13	Line printer 2
16	14	Synchronous modem 2
17	15	Terminal 18
20	16	Cassette drive 1
21	17	Cassette drive 2
22	18	Versatec on DMA printer/plotter 1
23	19	Versatec on DMA printer/plotter 2
24	20	Tektronix display
25	21	Magnetic tape 1 unit 2
26	22	Synchronous modem 5
27	23	Synchronous modem 6
30	24	Synchronous modem 3
31	25	Synchronous modem 4
32	26	Magnetic tape 2 unit 0

33	27	Magnetic tape 1 unit 3
34	28	Magnetic tape 2 unit 1
35	29	Card Punch 3
36	30	CDC link
37	31	Not used
40	32	Magnetic tape 1 unit 0
41	33	Magnetic tape 1 unit 1
42-47	34-39	Terminals 3-8
50	40	Card punch 1
51	41	Card punch 2
52-57	42-47	Terminals 19-24
60-67	48-55	Terminals 9-16
70-77	56-63	Terminals 25-32

*Logical Device Numbers 400-477:*

400-437	CAMAC interrupts or special process interface
440	Direct task level 6
441	Direct task level 7
442	Direct task level 8
443	Direct task level 9
450-467	CONNECT device

*Logical Device Numbers 500-577:*

500	Internal device for error message RT program
501	Semaphore for segment transfer
502	Disk 10Mb 1 datafield
503	RT loader semaphore
504	General lock for file system
505	User-file-buffer lock
506	Object-file-buffer lock
507	RT-open-file-table lock
510	Device buffer 1 lock (disk 1)
511	Disk 10Mb 1, unit 0, R-bit-file-buffer lock
512	Disk 10Mb 1, unit 0, F-bit-file-buffer lock
513	Disk 10Mb 1, unit 0, R-directory lock
514	Disk 10Mb 1, unit 0, F-directory lock
515	DF1, file-transfer lock for disk 1, drum 1, drum 2, disk 2
516	DF2, open-file monitor call from RT program datafield
517	RTFIL semaphore
520	Device buffer 2 lock
521	Get device buffer lock
522	Disk 10Mb 1, unit 1, R-directory lock
523	Disk 10Mb 1, unit 1, F-directory lock
524	Disk 10Mb 1, unit 1, R-bit-file-buffer lock
525	Disk 10Mb 1, unit 1, F-bit-file-buffer lock
526	DF3, file-transfer lock for magnetic tape 1 unit 0, magnetic tape 1 unit 1
527	Spooling queue semaphore
530	Accounting semaphore
531	CDC link monitor call datafield
532	Semaphore for spooling device 4
533	Semaphore for spooling device 4
534	Semaphore for spooling device 5

535	Semaphore for spooling device 5
536	Semaphore for spooling device 6
537	Semaphore for spooling device 6
540	Internal Device Remote Batch IBM
541	Internal Device Remote Batch UNIVAC
542	Internal Device Remote Batch Honeywell Bull
543	Internal Device Remote Batch CDC
544	Drum 2, datafield
545	Drum 2, bit-file-buffer lock
546	Drum 2, directory lock
547	Drum 2, device buffer lock
550	Drum 1, datafield
551	Drum 1, bit-file-buffer lock
552	Drum 1, directory lock
553	Drum 1, device buffer lock
554	Disk 10Mb 1, unit 3, R-bit-file-buffer lock
555	Disk 10Mb 1, unit 3, F-bit-file-buffer lock
556	Disk 10Mb 1, unit 3, R-directory lock
557	Disk 10Mb 1, unit 3, F-directory lock
560	Magnetic tape 1, datafield
561	Magnetic tape 1, unit 1, directory lock
562	Magnetic tape 2, unit 0, directory lock
563	Magnetic tape 2, unit 0, device buffer lock
564	Magnetic tape 1, unit 1, device buffer lock
565	Magnetic tape 1, unit 2, directory lock
566	Magnetic tape 1, unit 2, device buffer lock
567	CDC link data field
570	Disk 10Mb 1, unit 2, R-directory lock
571	Disk 10Mb 1, unit 2, F-directory lock
572	Disk 10Mb 1, unit 2, R-bit-file-buffer lock
573	Disk 10Mb 1, unit 2, F-bit-file-buffer lock
574	Monitor call datafield for cassette
575	Cassette data field
576	DF5, monitor call data field for Versatec
577	Versatec data field

*Logical Device Numbers 1000 - 1077:*

1000	Floppy disk 1, unit 0, I/O datafield
1001	Floppy disk 1, unit 1, I/O datafield
1002	Floppy disk 1, unit 2, I/O datafield
1003	Floppy disk 2, unit 0, I/O datafield
1004	Floppy disk 2, unit 1, I/O datafield
1005	Floppy disk 2, unit 2, I/O datafield
1006	Hasp DMA 1, I/O datafield
1007	Hasp DMA 2, I/O datafield
1010	Hasp DMA 3, I/O datafield
1011	Hasp DMA 4, I/O datafield
1012	Hasp DMA 5, I/O datafield
1013	Hasp DMA 6, I/O datafield
1014	Line printer 3, I/O datafield
1015	Line printer 4, I/O datafield
1040 - 1077	Terminals 33 - 64

*Logical Device Numbers 1100-1177:*

1100	Big disk/big cartridge disk 1, datafield
1101	Big disk/big cartridge disk 1, unit 0, directory lock
1102	Big disk/big cartridge disk 1, unit 0, bit-file-buffer lock
1103	NORD-50 datafield
1104	Disk 10Mb 2, datafield
1105	Disk 10Mb 2, unit 0, R-directory lock
1106	Disk 10Mb 2, unit 0, F-directory lock
1107	Disk 10Mb 2, unit 0, R-bit-file-buffer lock
1110	Disk 10Mb 2, unit 0, F-bit-file-buffer lock
1111	Magnetic tape 2 datafield
1112	Magnetic tape 2, unit 0, directory lock
1113	Magnetic tape 2, unit 0, device buffer lock
1114	Magnetic tape 2, unit 1, directory lock
1115	Magnetic tape 2, unit 1, device buffer lock
1116	Monitor call datafield for magnetic tape 2
1117	Big disk/big cartridge disk 1, unit 1, directory lock
1120	Big disk/big cartridge disk 1, unit 1, bit-file-buffer lock
1121	Big disk/big cartridge disk 1, unit 2, directory lock
1122	Big disk/big cartridge disk 1, unit 2, bit-file-buffer lock
1123	Big disk/big cartridge disk 1, unit 3, directory lock
1124	Big disk/big cartridge disk 1, unit 3, bit-file-buffer lock
1125	Versatec controller 2 datanet
1126	Monitor call datafield for Versatec controller 2
1127	Magnetic tape 2, unit 3, directory lock
1130	Disk 10Mb 2, unit 1, R-directory lock
1131	Disk 10Mb 2, unit 1, F-directory lock
1132	Disk 10Mb 2, unit 1, R-bit-file lock
1133	Disk 10Mb 2, unit 1, F-bit-file lock
1134	Disk 10Mb 2, device buffer lock
1135	Device buffer lock
1136	Semaphore for spooling device 1
1137	Semaphore for spooling device 1
1140	Semaphore for spooling device 2
1141	Semaphore for spooling device 2
1142	Spooling system semaphore
1143	Spooling system semaphore
1144	Spooling system semaphore
1145	Floppy disk 1, datafield
1146	Monitor call datafield for floppy disk 1
1147	Floppy disk 1, device buffer lock
1150	Floppy disk 1, unit 0, directory table lock
1151	Floppy disk 1, unit 0, bit-file-buffer lock
1152	Floppy disk 1, unit 1, directory table lock
1153	Floppy disk 1, unit 1, bit-file-buffer lock
1154	Floppy disk 1, unit 2, directory table lock
1155	Floppy disk 1, unit 2, bit-file-buffer lock
1156	Floppy disk 2, datafield
1157	Monitor call datafield for floppy disk 2
1160	Floppy disk 2, device buffer lock
1161	Floppy disk 2, unit 0, directory table lock
1162	Floppy disk 2, unit 0, bit-file-buffer lock
1163	Floppy disk 2, unit 1, directory table lock
1164	Floppy disk 2, unit 1, bit-file-buffer lock
1165	Floppy disk 2, unit 2, directory table lock
1166	Floppy disk 2, unit 2, bit-file-buffer lock

1167	Line printer 1, datafield
1170	Monitor call datafield for line printer 1
1171	Not used
1172	Not used
1173	Semaphore for spooling device 3
1174	Semaphore for spooling device 3
1175	Line printer 2, datafield
1176	Monitor call datafield lock for line printer 2
1177	Not used

*Logical Device Numbers 1200 - 1277:*

1200	NORD TPS system semaphore
1201	DMAC semaphore
1202	RT-PROGRAM-LOG semaphore
1203	Histogram commands lock
1204	SINTRAN-SERVICE-PROGRAM command lock
1205	Mail system lock
1206	Terminal 1, datafield
1207	Big disk/big cartridge disk 2, datafield
1210	Internal device 1, datafield
1211	Monitor call datafield for internal device 1
1212	Internal device 2, datafield
1213	Monitor call datafield for internal device 2
1214	Internal device 3, datafield
1215	Monitor call datafield for internal device 3
1216	Internal device 4, datafield
1217	Monitor call datafield for internal device 4
1220	Internal device 5, datafield
1221	Monitor call datafield for internal device 5
1222	Accounting semaphore
1223	GRAFS semaphore
1224 - 1227	Not used
1230	Device buffer lock
1231	Device buffer lock
1232	Device buffer lock
1233	Device buffer lock
1234	Device buffer lock
1235	Device buffer lock
1236	Batch process 1, datafield
1237	Batch process 1, internal device
1240	Batch process 2, datafield
1241	Batch process 2, internal device
1242	Batch process 3, datafield
1243	Batch process 3, internal device
1244	Batch process 4, datafield
1245	Batch process 4, internal device
1246	Batch process 5, datafield
1247	Batch process 5, internal device
1250	Batch process 6, datafield
1251	Batch process 6, internal device
1252	Batch process 7, datafield
1253	Batch process 7, internal device
1254	Batch process 8, datafield
1255	Batch process 8, internal device
1256	Batch process 9, datafield

1257	Batch process 9, internal device
1260	Batch process 10, datafield
1261	Batch process 10, internal device
1262	Spooling device 7, semaphore
1263	Spooling device 7, semaphore
1264	Spooling device 8, semaphore
1265	Spooling device 8, semaphore
1266	Spooling device 9, semaphore
1267	Spooling device 9, semaphore
1270	Spooling device 10, semaphore
1271	Spooling device 10, semaphore
1272	Monitor call datafield for internal device 1
1273	Monitor call datafield for internal device 2
1274	Monitor call datafield for internal device 3
1275	Monitor call datafield for internal device 4
1276	Monitor call datafield for internal device 5
1277	Not used

*Logical Device Numbers 1300 - 1377:*

1300	Device buffer lock
1301	Device buffer lock
1302	Device buffer lock
1303	Hasp DMA 1, datafield
1304	Hasp DMA 1, datafield
1305	Monitor call datafield for Hasp DMA 1
1306	Monitor call datafield for Hasp DMA 1
1307	Hasp DMA 2, datafield
1310	Hasp DMA 2, datafield
1311	Monitor call datafield for Hasp DMA 2
1312	Monitor call datafield for Hasp DMA 2
1313	Hasp DMA 3, datafield
1314	Hasp DMA 3, datafield
1315	Monitor call datafield for Hasp DMA 3
1316	Monitor call datafield for Hasp DMA 3
1317	Hasp DMA 4, datafield
1320	Hasp DMA 4, datafield
1321	Monitor call datafield for Hasp DMA 4
1322	Monitor call datafield for Hasp DMA 4
1323	Hasp DMA 5, datafield
1324	Hasp DMA 5, datafield
1325	Monitor call datafield for Hasp DMA 5
1326	Monitor call datafield for Hasp DMA 5
1327	Hasp DMA 6, datafield
1330	Hasp DMA 6, datafield
1331	Monitor call datafield for Hasp DMA 6
1332	Monitor call datafield for Hasp DMA 6
1333	Big disk/big cartridge disk 2, unit 0, directory table lock
1334	Big disk/big cartridge disk 2, unit 0, bit-file-buffer lock
1335	Big disk/big cartridge disk 2, unit 1, directory table lock
1336	Big disk/big cartridge disk 2, unit 1, bit-file-buffer lock
1337	Big disk/big cartridge disk 2, unit 2, directory table lock
1340	Big disk/big cartridge disk 2, unit 2, bit-file-buffer lock
1341	Big disk/big cartridge disk 2, unit 3, directory table lock
1342	Big disk/big cartridge disk 2, unit 3, bit-file- buffer lock

1343	Line printer 3, datafield
1344	Monitor call datafield for line printer 3
1345	Line printer 4, datafield
1346	Monitor call datafield for line printer 4
1347	NORD-50 2, datafield
1350	NORD-50 3, datafield
1351	NORD-50 4, datafield
1352	RT-PROGRAM-LOG command lock
1360	HDLC DMA, link 1, input; synchronous modem 1 for HDLC interface input/output
1361	HDLC DMA, link 1, output
1362	HDLC DMA, link 2, input; synchronous modem 2 for HDLC interface input/output
1364	HDLC DMA, link 3, input; synchronous modem 3 for HDLC interface input/output
1366	HDLC DMA, link 4, input; synchronous modem 4 for HDLC interface input/output
1373	HDLC DMA, link 6, output; synchronous modem 6 for HDLC interface output.
1374	X21 logical number 1
1375	X21 logical number 2
1376	X21 logical number 3
1377	X21 logical number 4



## APPENDIX D

### ERROR MESSAGES

This appendix documents SINTRAN run-time and file system errors. Error messages originating in the subsystems are documented in the respective subsystem documentation; this also includes FORTRAN run-time errors, BASIC run-time errors, etc.

#### D.1 SINTRAN MONITOR

##### D.1.1 Run-Time Errors

Most run-time errors cause the current RT program to be aborted and the error message

```
aa.bb.cc ERROR nn IN rr AT ll ; tttt
```

```
xx yy
```

is printed.

If the error occurs in a background program, the error message is written on the corresponding terminal. For RT programs, the error message is written on the error message terminal (usually terminal 1).

The meaning of the parameters is as follows:

aa.bb.cc            Time when the error message was printed.

aa    hours

bb    minutes

cc    seconds

nn            Error number (see following pages).

rr            Name of RT program.

ll            Octal address where the error occurred.

tttt           Explanatory text.

xx, yy        Numbers giving additional information about the error. One or both numbers can be omitted. (See following pages.)

**Example:**

```
@01.43.32 ERROR 14 IN RTP 1 AT 114721;
```

```
OUTSIDE SEGMENT BOUNDS
```

In the case of a segment transfer error, an additional message TRANSF! is given.

## D.1.2 Run-Time Error Codes

<i>Error Code</i>	<i>Meaning</i>	<i>xx</i>	<i>yy</i>	<i>Program Aborted</i>
00	Illegal monitor call			yes
01	Bad RT program address			yes
02	Wrong priority in PRIOR			yes
03	Bad memory page	page no.		
04	Internal interrupt on direct task level	level	bit no.	
06	Batch input error	error no.		yes
07	Batch output error	error no.		yes
08	Batch system error	error no.	L register	yes
09	Illegal parameter in CLOCK			yes
10	Illegal parameter in ABSET			yes
11	Illegal parameter in UPDAT			yes
12	Illegal time parameters			yes
13	Page fault for non-demand	page no.		yes
14	Outside segment bounds	page no.		yes
15	Illegal segment number	segment no.		yes
16	Segment not loaded	segment no.		yes
17	Fixing demand	segment no.		yes
18	Too many fixed pages	segment no.		yes
19	Too big segment	segment no.		yes
20	Disk/drum transfer error	hardware device no.	unit	no (yes if segment transfer)
21	Disk/drum transfer error	disk address	hardware status	no

<i>Error Code</i>	<i>Meaning</i>	<i>xx</i>	<i>yy</i>	<i>Program Aborted</i>
22	False interrupt	level	ident code	no
23	Device error	hardware device no.	hardware status	no
25	Already fixed	segment no.		yes
26	Device time-out	hardware device no.	unit no.	no
27	Illegal parameter in CONCT			yes
28	Space not available	segment no.		yes
29	MON 64 and MON 65	error no.	(see NORD File System)	yes
30	Divide by zero			yes
31	Permit violation			yes
32	Ring violation			yes
33	HDLC driver, fatal error			yes
34	Illegal instruction			yes
35	REENTRANT-FTN stack error			yes
36	Privileged instruction			yes
37	IOX error	address	level	no
38	Memory parity error	PEA reg.	PES reg.	yes
39	Memory out of range	PEA reg.	PES reg.	yes
40	Power fail			no
41	Illegal error code in ERRMON			yes
42	Overlapping segments	segments		yes
44	Corrupted memory error	PEA reg.	PES reg.	no
45	Not demand segments			yes
46	XMSG fatal error, internal error or inconsistency	XMSG error code	physical address	yes

<i>Error Code</i>	<i>Meaning</i>	<i>xx</i>	<i>yy</i>	<i>Program Aborted</i>
47	XMSG user error	calling level		yes
48	False BEX interrupt			
49	Remote power fail interrupt			
50-69	User defined error (MON 142)	error no.	suberror no.	no
70	BEX parity error			
90	FORTTRAN run-time error	error no.		no
91	FORTTRAN I/O error	error no.		no
100	FTN library error			

## D.2 SINTRAN III FILE SYSTEM

### D.2.1 Error Codes Returned from Monitor Calls

<i>Error Code:</i>		<i>Meaning:</i>
Octal	Decimal	
000	000	Not used
001	001	Not used
002	002	Bad file number
003	003	End of file
004	004	Card Reader Error (card read)
005	005	Device not reserved
006	006	Not used
007	007	Card Reader Error (card not read)
010	008	Not used
011	009	Not used
012	010	End of device (time-out)
013	011	Not used
014	012	Not used
015	013	Not used
016	014	Not used
017	015	Not used
020	016	Not used
021	017	Illegal character in parameter
022	018	No such page
023	019	Not decimal number
024	020	Not octal number
025	021	You are not authorized to do this
026	022	Directory not entered
027	023	Ambiguous directory name
030	024	No such device name
031	025	Ambiguous device name
032	026	Directory entered
033	027	No such logical unit
034	028	Unit occupied
035	029	Master block transfer error
036	030	Bit file transfer error
037	031	No more tracks available
040	032	Directory not on specified unit
041	033	Files opened on this directory
042	034	Main directory not last one released
043	035	No main directory
044	036	Too long parameter
045	037	Ambiguous user name
046	038	No such user name
047	039	No such user name in main directory
050	040	Attempt to create too many users
051	041	User already exists
052	042	User has files
053	043	User is entered
054	044	Not so much space unreserved in directory
055	045	Reserved space already used
056	046	No such file name
057	047	Ambiguous file name
060	048	Wrong password

<i>Error Code:</i>	<i>Meaning:</i>
Octal	Decimal
061	049 User already entered
062	050 No user entered
063	051 Friend already exists
064	052 No such friend
065	053 Attempt to create too many friends
066	054 Attempt to create yourself as friend
067	055 Continuous space not available
070	056 Not directory access
071	057 Space not available to expand file
072	058 Space already allocated
073	059 No space in default directories
074	060 No such file version
075	061 No more pages available for this user
076	062 File already exists
077	063 Attempt to create too many files
100	064 Outside device limits
101	065 No previous version
102	066 File not continuous
103	067 File type already defined
104	068 No such access code
105	069 File already opened
106	070 Not write access
107	071 Attempt to open too many files
110	072 Not write and append access
111	073 Not read access
112	074 Not read, write and common access
113	075 Not read and write access
114	076 Not read and common access
115	077 File reserved by another user
116	078 File already opened for write by you
117	079 No such user index
120	080 Not append access
121	081 Attempt to open too many mass-storage files
122	082 Attempt to open too many files
123	083 Not opened for sequential write
124	084 Not opened for sequential read
125	085 Not opened for random write
126	086 Not opened for random read
127	087 File number out of range
130	088 File number already used
131	089 No more buffer space
132	090 No file opened with this number
133	091 Not mass-storage file
134	092 File used for write
135	093 File used for read
136	094 File only opened for sequential read or write
137	095 No scratch file opened
140	096 File not reserved by you
141	097 Transfer error
142	098 File already reserved
143	099 No such block
144	100 Source and destination equal
145	101 Illegal on tape device
146	102 End of tape

<i>Error Code:</i>		<i>Meaning:</i>
Octal	Decimal	
147	103	Device unit reserved for special use
150	104	Not random access on tape files
151	105	Not last file on tape
152	106	Not tape device
153	107	Illegal address reference in monitor call
154	108	Source empty
155	109	File already open by another user
156	110	File already open for write by another user
157	111	Missing parameter
160	112	Two pages must be left unreserved
161	113	No answer from remote computer
162	114	Device cannot be reserved
163	115	Overflow in read
164	116	DMA error
165	117	Bad data block
166	118	CONTROL MODUS word error
167	119	Parity error
170	120	LRC error
171	121	Device error (read-last-status to get status)
172	122	No device buffer available
173	123	Illegal mass-storage unit number
174	124	Illegal parameter
175	125	Writeprotect violation
176	126	Error detected by read after write
177	127	No EOF mark found
200	128	Cassette not in position
201	129	Illegal function code
202	130	Time-out (no data block found)
203	131	Paper fault
204	132	Device not ready
205	133	Device already reserved
206	134	Not peripheral file
207	135	No such queue entry
210	136	Not so much space left
211	137	No spooling for this device
212	138	No such queue
213	139	Queue empty
214	140	Queue full
215	141	Not last used by you
216	142	No such channel name
217	143	No remote connection
220	144	Illegal channel
221	145	Channel already reserved on remote computer
222	146	No remote file processor
223	147	Formatting error
224	148	Incompatible device sizes
225	149	Remote processor not available
226	150	Tape format error
227	151	Block count error
230	152	Volume not on specified unit
231	153	Not deleted record
232	154	Device error
233	155	Error in object entry

*Error Code:**Meaning:*

Octal	Decimal	
-------	---------	--

234	156	Odd number of bytes (right byte in last word insignificant)
235	157	Error in backspace/forward space print
236	148	Block format error
237	159	Overflow in write
240	160	Illegal device type
241	161	Segment not contiguously fixed
242	162	Segment not fixed
243	163	Approaching end of accounting file
244	164	End of accounting file encountered
245	165	No more unused spooling files available
246	166	Inconsistent directory
300-377		Reserved for special use



## APPENDIX E

## STANDARD PERIPHERAL FILE NAMES

ND No.	Description	Peripheral File Name	Notes
202, 204 — 228	Terminals	TERMINAL	1. The user's terminal in background. 2. Terminal can be PRINTER.
252, 254	Intercomputer link	CHANNEL —0 CHANNEL —1 ... CHANNEL —15  L1-CH-0 ... L1-CH-15 L2-CH-0 ... L2-CH-15 ...	Only one link    Two or more links   Links with background programs are not usually included.
301, 302	Paper Tape Reader	TAPE-READER	Suffix "—1", "—2", etc. is used if more than one device.
303	Page Tape Punch	TAPE-PUNCH	See ND301.
305, etc.	Floppy Disk	FLOPPY—0 FLOPPY—1	These names only work with one controller
400, etc.	Card Reader	... CARD-READER	See ND301
430, 431, etc.	Line Printer	LINE-PRINTER	See ND301
414, 415, 417	Matrix Printer	PRINTER	See ND301
420	Card Punch	CARD-PUNCH	See ND301
515, etc.	Magnetic Tape	MAG-TAPE-0 MAG-TAPE-1	These names only work with one controller
603, 604, 605, 606	Versatec Printer Plotter	LINE-PRINTER-1 LINE-PRINTER-2 ...  VERSATEC-1 VERSATEC-2 ...	If no other line printer on the system.   If another line printer on the system.

## APPENDIX F

## STANDARD NAMES OF MASS STORAGE DEVICES

DISC-10MB-1	Cartridge disk controller 1
DISC-10MB-2	Cartridge disk controller 2
DISC-30MB-1	Big cartridge disk controller 1
DISC-30MB-2	Bit cartridge disk controller 2
DISC-33MB-1	33 Megabytes disk controller 1
DISC-33MB-2	33 Megabytes disk controller 2
DISC-38MB-1	38 Megabytes disk controller 1
DISC-38MB-2	38 Megabytes disk controller 2
DISC-60MB-1	60 Megabytes disk controller 1
DISC-60MB-2	60 Megabytes disk controller 2
DISC-66MB-1	66 Megabytes disk controller 1
DISC-66MB-2	66 Megabytes disk controller 2
DISC-75MB-1	75 Megabytes disk controller 1
DISC-75MB-2	75 Megabytes disk controller 2
DISC-90MB-1	90 Megabytes disk controller 1
DISC-90MB-2	90 Megabytes disk controller 2
DISC-288MB-1	288 Megabytes disk controller 1
DISC-288MB-2	288 Megabytes disk controller 2
DISC-2-75MB-1	
DISC-2-75MB-2	
DISC-3-75MB-1	
DISC-3-75MB-2	
DRUM-1	Drum controller 1
DRUM-2	Drum controller 2
MAG-TAPE-1	Magnetic tape controller 1
MAG-TAPE-2	Magnetic tape controller 2
FLOPPY-DISC-1	Floppy disk controller 1
FLOPPY-DISC-2	Floppy disk controller 2

## APPENDIX G

## ASCII CHARACTER SET

Byte Position				Byte Position			
CHAR	Left	Right	Dec.	CHAR	Left	Right	Dec.
NUL	000000	000000	0	0	030000	000060	48
SOH	000400	000001	1	1	030400	000061	49
STX	001000	000002	2	2	031000	000062	50
ETX	001400	000003	3	3	031400	000063	51
EOT	002000	000004	4	4	032000	000064	52
ENQ	002400	000005	5	5	032400	000065	53
ACK	003000	000006	6	6	033000	000066	54
BEL	003400	000007	7	7	033400	000067	55
BS	004000	000010	8	8	034000	000070	56
HT	004400	000011	9	9	034400	000071	57
LF	005000	000012	10	:	035000	000072	58
VT	005400	000013	11	;	035400	000073	59
FF	006000	000014	12	<	036000	000074	60
CR	006400	000015	13	=	036400	000075	61
SO	007000	000016	14	>	037000	000076	62
SI	007400	000017	15	?	037400	000077	63
DLE	010000	000020	16	@	040000	000100	64
DC1	010400	000021	17	A	040400	000101	65
DC2	011000	000022	18	B	041000	000102	66
DC3	011400	000023	19	C	041400	000103	67
DC4	012000	000024	20	D	042000	000104	68
NAK	012400	000025	21	E	042400	000105	69
SYN	013000	000026	22	F	043000	000106	70
ETB	013400	000027	23	G	043400	000107	71
CAN	014000	000030	24	H	044000	000110	72
EM	014400	000031	25	I	044400	000111	73
SUB	015000	000032	26	J	045000	000112	74
ESC	015400	000033	27	K	045400	000113	75
FS	016000	000034	28	L	046000	000114	76
GS	016400	000035	29	M	046400	000115	77
RS	017000	000036	30	N	047000	000116	78
US	017400	000037	31	O	047400	000117	79
SPACE	020000	000040	32	P	050000	000120	80
!	020400	000041	33	Q	050400	000121	81
"	021000	000042	34	R	051000	000122	82
#	021400	000043	35	S	051400	000123	83
\$	022000	000044	36	T	052000	000124	84
%	022400	000045	37	U	052400	000125	85
&	023000	000046	38	V	053000	000126	86
'	023400	000047	39	W	053400	000127	87
(	024000	000050	40	X	054000	000130	88
)	024400	000051	41	Y	054400	000131	89
*	025000	000052	42	Z	055000	000132	90
+	025400	000053	43	[	055400	000133	91
,	026000	000054	44	\	056000	000134	92
-	026400	000055	45	]	056400	000135	93
.	027000	000056	46	^	057000	000136	94
_	027400	000057	47				

CHAR	Byte Position		Dec.	CHAR	Byte Position		Dec.
	Left	Right			Left	Right	
—	057400	000137	95	o	067400	000157	111
	060000	000140	96	p	070000	000160	112
a	060400	000141	97	q	070400	000161	113
b	061000	000142	98	r	071000	000162	114
c	061400	000143	99	s	071400	000163	115
d	062000	000144	100	t	072000	000164	116
e	062400	000145	101	u	072400	000165	117
f	063000	000146	102	v	073000	000166	118
g	063400	000147	103	w	073400	000167	119
h	064000	000150	104	x	074000	000170	120
i	064400	000151	105	y	074400	000171	121
j	065000	000152	106	z	075000	000172	122
k	065400	000153	107		075400	000173	123
l	066000	000154	108		076000	000174	124
m	066400	000155	109		076400	000175	125
n	067000	000156	110		077000	000176	126
				DEL	077400	000177	127

## APPENDIX H

### HARDWARE STATUS FOR MAGNETIC TAPE, CASSETTE TAPE, VERSATEC AND FLOPPY DISK.

This is the status word returned by  
 @DEVICE-FUNCTION <peripheral file name>  
 FUNCTIONS: READ-STATUS  
               READ-LAST-STATUS  
 and by MAGTP (MON 144) function codes 20 and 24.  
 The specified condition is true if the bit is set in the status word.

#### H.1 Status word for Tandberg, Pertec and STC magnetic tape units.

- bit 0: tape on line
  - 1: write enable ring present
  - 2: tape standing onload point
- 3: CRC error/fatal error
- 4: set if any of bits 5, 6, 7, 8, 9, 11 or 12 are set
- 5: control or modus word error; trying to write on protected tape, trying to reverse tape at load point, tape unit not on-line etc.; action is inhibited
- 6: bad data block; an error is detected
- 7: end of file is detected
- 8: the search character is detected
- 9: end of tape is detected; resetting this bit depends on the model.  
     Tandberg, STC: the bit remains set if carrying out a function after EOT  
     Pertec: the bit is cleared if carrying out a function after EOT
- 10: word counter is not zero
- 11: DMA error
- 12: overflow (in read)
- 13: tape busy or formatter busy
- 14: LRC error/software error
- 15: interrupt when formatter is ready

## H.2 Status word for Hewlett-Packard magnetic tape units

- bit 0: ready interrupt enabled (cleared by the interrupt)  
1: error interrupt enabled (cleared by the interrupt)  
2: device active
- 3: device ready for transfer  
4: set if any of bits 6, 9, 10, 11 or 12 are set or if a reverse command is given with tape at load point  
5: write enable ring present
- 6: LRC error  
7: EOF detected  
8: load point (the unit remains in this state also after the first forward command after load point is detected)
- 9: EOT detected  
10: parity error  
11: DMA error
- 12: overflow in read  
13: density select: 1 = 800 BPI  
14: magnetic tape unit ready (selected, on-line and not rewinding)
- 15: bit 15 is loaded by the previous control word

### H.3 Status word for Phillips cassette

bit 0: ready for transfer, interrupt is enabled

1: error interrupt enabled

2: device is active

3: device is ready for transfer

4: set if any of bits 0, 1 or 5 are set

5: write enable

6: cassette side indicator (A = 1, B = 0)

7: bit clock

8: read fail

9: sync fail

10: not used

11: not used

12: drive fail

13: write protection violation

14: beginning or end of tape

15: not used

## H.4 Status word for Versatec line printer/plotter

- bit 0: ready for transfer, interrupt enabled
- 1: error interrupt enabled
- 2: device active
  
- 3: device ready for transfer
- 4: set if bit 6 or 7 is set
- 5: not used
  
- 6: no paper
- 7: plotter not on-line
- 8: not used
  
- 9: not used
- 10: not used
- 11: not used
  
- 12: not used
- 13: plotter ready
- 14: not used
  
- 15: not used



## H.5 Status word for floppy disk

- bit 0: interrupt enabled
- 1: not used
- 2: device busy
- 3: device ready for transfer
- 4: set if any of bits 5, 8, 11, 12 or 14 are set
- 5: deleted record detected
- 6: read/write completed
- 7: seek completed
- 8: drive not ready
- 9: write protected
- 10: not used
- 11: address mismatch
- 12: CRC error
- 13: not used
- 14: data overrun
- 15: not used

## APPENDIX I

MONITOR CALLS AVAILABLE FROM FORTRAN ON  
ND-100 AND ND-500

Roman numerals in brackets refer to the notes which follow the table.

NAME	NUMBER	ARGUMENTS	TYPE	COMMENTS	ERRCODE SET
LEAVE	0			no return	
INCH	1	1. input unit	integer	Fortran unit number	yes
(iv)		<i>function value</i>	integer	(iii)	
OUTCH	2	1. output unit	integer	Fortran unit number	yes
(iv)		2. output char	integer	right-adjusted	
		<i>function value</i>	integer	(iii)	
ECHOM	3	1. device	integer	SINTRAN LDN	no
		2. strategy	integer	$0 \leq \text{strategy} \leq 7$	
		3. table	int array	for strategy $\leq 7$	
				8 words - ND-100	
				4 words - ND-500	
BRKM	4	1. device	integer	SINTRAN LDN	no
		2. strategy	integer	strategy	
		3. no. characters	integer	for strategy $\geq 3$	
		4. table	int array	for strategy = 7	
				8 words - ND-100	
				4 words - ND-500	
TIME	11	<i>function value</i>	integer*4	in basic time units	no
CIBUF	13	1. unit	integer	Fortran unit number	yes
COBUF	14	1. unit	integer	Fortran unit number	yes
MGTTY	16	1. unit	integer	Fortran unit number	yes
		2. <i>terminal type</i>	integer	returned terminal type	
MSTTY	17	1. unit	integer	Fortran unit number	yes
		2. terminal type	integer	terminal type	
LASTC	26	1. device	integer	SINTRAN LDN	no
		<i>function value</i>	integer	last character	
RTDSC	27	1. RT desc			
		address	integer		no
		2. RT desc	int array	26 <sub>8</sub> words minimum	
GETRT	30	<i>function value</i>	integer		no
EXIOX	31	1. register	integer	A-register ND-100	no
				I1-register ND-500	
		2. dev reg addr	integer		
MSG	32	1. message	character	(vi)	no
ALTON	33	1. page table no	integer	alt page table	no
				used to access	
ALTOF	34			COMMON (v)	no
				alt page table	
				used to access	
				COMMON (v)	
IOUT	35	1. radix	integer	2, 8, 10, 16	no
		2. value	integer		

NOWT	36	1. device	integer	SINTRAN LDN	no
		2. I/O flag	integer	= 0 input; = 1 output	
		3. wait flag	integer	= 0 I/O wait mode	
AIRDW	37	1. no channels	integer	= N	no
		2. channel nos	int*2 array	N 16-bit words	
		3. read values	int*2 array	N 16-bit words	
		4. error flag	integer		
SPCLO	40	1. unit	integer	Fortran unit number	yes
		2. text string	character	(vi)	
		3. no of copies	integer		
		4. print flag	integer		
ROBJE	41	1. unit	integer	Fortran unit number	no
		2. <i>buffer</i>	integer	any type except character, 32 <sub>10</sub> words minimum	
RUSER	44	1. user name	character	(vi)	no
		2. <i>buffer</i>	integer	any type except character, 32 <sub>10</sub> words minimum	
TERMO	52	1. device	integer	SINTRAN LDN	no
		2. mode	integer	terminal mode	
DLFI	54	1. file name	character	(vi)	yes
PASET	56	1-5 parameters	integer	set error parameters	no
PAGET	57	1-5 <i>parameters</i>	integer	get error parameters	no
RMAX	62	1. unit	integer	Fortran unit number	yes
		2. <i>no of bytes</i>	integer*4	number of bytes available for sequential read	
ERMSG	64	1. error number	integer	print error message	no
QERMS	65	1. error number	integer	program terminates	
ISIZE	66	1. unit	integer	Fortran unit number	no
		<i>function value</i>	integer	no. characters in input buffer (iii)	
OSIZE	67	1. unit	integer	Fortran unit number	no
		<i>function value</i>	integer	no. characters in output buffer (iii)	
COMND	70	1. command string	character	(vi)	no
DESCF	71	1. device	integer	SINTRAN LDN	no
EESCF	72	1. device	integer	SINTRAN LDN	no
SMAX	73	1. unit	integer	Fortran unit number	yes
		2. byte count	integer*4	max byte number	
SETBT	74	1. unit	integer	Fortran unit number	yes
		2. byte pointer	integer*4	first byte has no. 0	
REABT	75	1. unit	integer	Fortran unit number	yes
		2. <i>byte pointer</i>	integer*4	first byte has no. 0	
SETBS	76	1. unit	integer	Fortran unit number	yes
		2. block size	integer*4	size in words	
SETBL	77	1. unit	integer	Fortran unit number	yes
		2. block number	integer*4	first block has no. 0	
RT	100	1. RT-program	external	RT-description address	no
SET	101	1. RT-program	external	RT-description address	no
		2. no. of time units	integer		
		3. unit size	integer		
ABSET	102	1. RT-program	external	RT-description address	no
		2. seconds	integer		
		3. minutes	integer		
		4. hours	integer		

INTV	103	1. RT-program 2. no. of time units 3. unit size	external integer integer	RT-description address	no
HOLD	104	1. no. of time units 2. unit size	integer integer		no
ABORT	105	1. RT-program	external	RT-description address	no
CONCT	106	1. RT-program 2. device	external integer	RT-description address SINTRAN LDN	no
DSCNT	107	1. RT-program	external		no
PRIOR	110	1. RT-program 2. priority <i>function value</i>	external integer integer	RT-description address old priority	no
UPDAT	111	1-5 time	integer	set internal clock	no
CLADJ	112	1. no. of time units 2. basic time units	integer integer	adjust internal clock	no
CLOCK	113	1. <i>time</i>	int array	7 words	no
TUSED	114	<i>function value</i>	integer*4		no
FIX	115	1. segment no	integer		no
UNFIX	116	1. segment no	integer		no
RFILE	117	1. unit 2. return flag 3. memory addr 4. block no 5. length to read <i>function value</i>	integer integer array integer integer integer integer	Fortran unit number any type except character count in words	yes
WFILE	120	1. unit 2. return flag 3. memory addr 4. block no 5. length to read <i>function value</i>	integer integer array integer integer integer integer	Fortran unit number any type except character count in words	yes
WAITF	121	1. unit 2. return flag <i>function value</i>	integer integer integer	Fortran unit number	yes
RESRV	122	1. device 2. I/O flag 3. return flag <i>function value</i>	integer integer integer integer	SINTRAN LDN	no
RELES	123	1. device 2. I/O flag	integer integer	SINTRAN LDN	no
PRSRV	124	1. device 2. I/O flag 3. RT-program <i>function value</i>	integer integer external integer	SINTRAN LDN RT-description address	no
PRLS	125	1. device 2. I/O flag	integer integer	SINTRAN LDN	no
DSET	126	1. RT-program 2. delay	external integer*4		no
DABST	127	1. RT-program 2. time	external integer*4	RT-description address	no
DINTV	130	1. RT-program 2. time interval	external integer*4	RT-description address	no

ABSTR	131	1. device	integer	SINTRAN LDN	no
		2. function	integer		
		3. memory addr	integer*4	double integer	
		4. block addr	integer		
		5. no. blocks	integer		
MEXIT	133	1. segment no	integer	right byte only	no
RTEXT	134			no return	
RTWT	135				no
RTON	136	1. RT-program	external	RT-description address	no
RTOFF	137	1. RT-program	external	RT-description address	no
WHDEV	140	1. device	integer	SINTRAN LDN	no
		2. I/O flag	integer		
		<i>function value</i>	integer	= 0 device free ≠ 0 RT-description address of reserving program	
IOSET	141	1. device	integer	SINTRAN LDN	no
		2. I/O flag	integer	= 0 input; = 1 output	
		3. RT-description address	integer	must = 0	
		4. control flag	integer		
		<i>function value</i>	integer		
ERMON	142	1. error number	integer		no
		2. suberror no	integer		
RSIO	143	1. exec mode	integer	= 0 iterative = 1 mode = 2 batch SINTRAN LDN SINTRAN LDN	no
		2. input device	integer		
		3. output device	integer		
		4. user number	integer		
MAGTP	144	1. function	integer		no
		2. memory addr	array	any type except character	
		3. unit	integer	Fortran unit number	
		4. parameter 1	integer	dependent on function	
		5. parameter 2	integer	may not be required	
		<i>function value</i>	integer		
ACM	145	1. device	integer	SINTRAN LDN	no
		2. function	integer		
		3. memory addr	array		
		4. destination	array		
		5. word count	integer	number of words	
		<i>function value</i>	integer		
CAMAC	147	1. data	integer		no
		2. status	integer		
		3. crate	integer		
		4. station	integer		
		5. subaddress	integer		
		6. function	integer		
GL	150	1. value	integer		no
		2. crate	integer		
GRDTA	151	1. name	hollerith	RT-program name ending with apostrophe	no
		<i>function value</i>	integer		
IOXN	153	1. data	integer		no
		2. IOX code	integer		
ASSIG	154	1. device	integer	SINTRAN LDN	no
		2. graded LAM	integer		
		3. crate	integer		

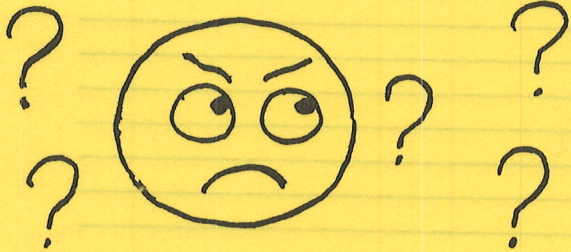
GRAPHIC	155	1-2 X,Y co-ord	integer			
		3. code	integer			no
		4. device	integer	SINTRAN LDN		
		5. function	integer			
		<i>function value</i>	integer			
ENTSG	157	1. segment	integer			no
		2. page table	integer			
		3. interrupt level	integer			
		4. entry point	integer			
FIXC	160	1. segment no	integer			no
		2. page no	integer	first physical page no		
INSTR	161	1. unit	integer	Fortran unit number		no
		2. <i>text</i>	character	starts on a word boundary		
		3. length	integer	length in bytes		
		4. end char	integer	right hand byte used		
		<i>function value</i>	integer			
OUTST	162	1. unit	integer	Fortran unit number		no
		2. text	character	starts on a word boundary		
		3. length	integer	length in bytes		
		<i>function value</i>	integer			
WSEG	164	1. segment no	integer			no
DIW	165	1. no of regs	integer	= N		no
		2. input regs	int*2 array	N 16-bit words		
		3. input values	int*2 array	N 16-bit words		
		4. error value	integer			
DOLW	166	1. no of regs	integer	= N		no
		2. registers	int*2 array	N 16-bit words		
		3. output values	int*2 array	N 16-bit words		
		4. masks	int*2 array	N 16-bit words		
		5. error value	integer			
REENT	167	1. segment no	integer			no

## NOTES:

- (i) In cases where SINTRAN III requires a LDN for a file the Fortran runtime routines generally accept a Fortran unit number. If there is no file opened with this unit number the Fortran runtime system assumes it is a SINTRAN LDN.  
 The user can convert from Fortran unit number to SINTRAN logical device number by using the function LDN. This takes a Fortran unit number as its single argument (integer\*2 on ND-100 and integer\*4 on ND-500) and returns the SINTRAN logical device number (integer\*2 on ND-100 and integer\*4 on ND-500).  
 If there is no open file for the Fortran unit the argument is returned unchanged.  
 Fortran unit number and logical device number are identical on the ND-500.
- (ii) Integer in the data type column means default integer type, integer\*2 on ND-100 and integer\*4 on ND-500.
- (iii) Function value. These monitor calls are made as functions, eg.  
         ICCHAR = INCH(UNIT)  
 In some cases, as in this example, the required data is returned as the function value. When an error occurs the function value returns the error code with the sign bit set to 1.
- (iv) The names of the monitor calls corresponding to these routines are  
         INCH - INBT  
         OUTCH - OUTBT
- (v) Available on ND-100 only.
- (vi) Must start on a word boundary (ND-100 only) and end with an apostrophe.
- (vii) In the list of arguments, input parameters are in normal type *and return values in italics*.



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

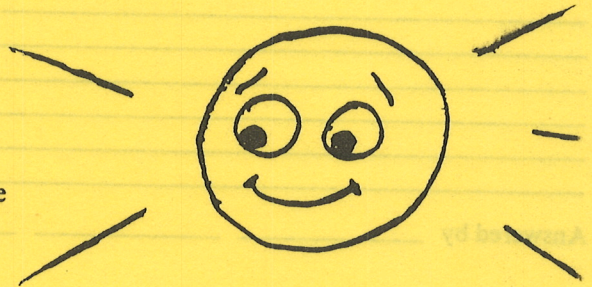


Are you frustrated because of unclear information in this manual? Do you have trouble finding things? Why don't you join the Reader's Club and send us a note? You will receive a membership card - and an answer to your comments.

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: SINTRAN III Reference Manual

Manual number: ND-60.128.02

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

---

---

---

---

---

---

---

---

---

---

Do you have suggestions for improving this manual?

---

---

---

---

---

---

---

---

---

---

Your name: \_\_\_\_\_ Date: \_\_\_\_\_

Company: \_\_\_\_\_ Position: \_\_\_\_\_

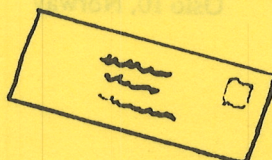
Address: \_\_\_\_\_

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! \*\*\*\*\*

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 questions.

Please let us know if you:

- \* find errors
- \* cannot understand information
- \* cannot find information
- \* find useless information

Do you think we could improve the manual by rearranging the contents?

Answered by

Date

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

Manual number: ND-80.138.02

Manual name: SINTRAM III Reference Manual

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

Do you have suggestions for improving this manual?

I I  
I I  
I I

Date:

Position:

Norsk Data A.S.

Documentation Department

P.O. Box 4, Lindeberg Gård

Oslo 10, Norway

Send to: Norsk Data A.S.

Documentation Department

P.O. Box 4, Lindeberg Gård

Oslo 10, Norway



**– we make bits for the future**

NORSK DATA A.S BOX 4 LINDEBERG GÅRD OSLO 10 NORWAY PHONE: 30 90 30 TELEX: 18661