RCSL No: 31-D688

Edition: March 1983

Author: Niels Møller Jørgensen
Flemming Biggas

Title:

PRIMO (4th Edition)
User's Guide/Reference Manual/Installation Guide

RCSL No 42-i 2164

A/S REGNECENTRALEN
af 1979

Keywords:

RC8000, operating system, printer, punch, paper tape reader, card reader, teletype, backing storage.

Abstract:

PRIMO is a restricted version of a general File Router. PRIMO supports transports of data on a file level between backing storage and external devices.

This manual contains information of interest for the application programmer, the operator, and the system staff.

(84 printed pages)

Printed by A/S Regnecentralen af 1979, Copenhagen

RCSL No 42-i 2164

FOREWORD

First edition: RCSL No 31-D504.

Second edition: RCSL No 31-D571.

Third edition: RCSL No 31-D677.
This manual describes revision 3 of the file router PRIMO.

Compared to revision 2 the following changes have been made:

1) A number of new operator functions for printer handling have been implemented.

2) The performance has been improved.

The above mentioned changes have increased PRIMO's demand for core substantially compared to revision 2.

This manual replaces the description at revision 2 (RCSL No 31-D571, May 1979).

Flemming Biggas
A/S REGNECENTRALEN af 1979, July 1982

Fourth edition: RCSL No 31-D688.
This manual describes revision 4 of the file router PRIMO.

Compared to revision 3 the following extensions have been made:

1) PRIMO is now able to handle transports from bs to IBM 3270 printers.

2) The operator module has been extended allowing the main operator to handle all devices known to PRIMO. As a con-

sequence the syntax and layout for operator communication have been extended and a new command 'display' has been implemented.

Correction lines in left margin indicate extensions and changes compared to revision 3.

Flemming Biggas
A/S REGNECENTRALEN af 1979, January 1983

TABLE OF CONTENTS PAGE

TABLE OF CONTENTS (continued) PAGE

TABLE OF CONTENTS (continued) PAGE

1. INTRODUCTION 1.

This manual contains information about the file router PRIMO.

Relevant information may be found by people, who are going to use, install, generate or work as an operator for PRIMO.

PRIMO is a restricted version of a general File Router module. It aims specifically at support of character oriented devices like printers and paper tape readers, and its main advantage is that it allows any userjob (whether executed under s, SOS or BOSS) to ask for files to be moved between a character oriented device and a backing storage area. This ensures that the userjob is not delayed by a slow external device, because the job may proceed concurrently with the device handling. If needed, the userjob may later ask for the state of the transport.

The interface formats to PRIMO are a subset of the formats for the File Router. This means that the information handed over from an application program to PRIMO may, without modification also be handed over to the File Router. In the information returned, the File Router will presumably deliver more information than is the case with PRIMO. This may call for reprogramming of programs communicating with the File Router on the message answer level.

The external ALGOL procedure, described in chapter 5, hides these modfications, the price for this being that only certain, most common, functions can be utilized.

This manual contains the following information:

Chapter 2 describes some terms used within PRIMO and the File Router.

Chapter 3 describes the meaning of the information exchanged between PRIMO and an application program.

Chapter 4 describes the formats of the information exchanged between PRIMO and an application program.

Chapter 5 describes an ALGOL procedure which may be used by an application programmer to define transports, to ask for their state and to wait for their completion.

Chapter 6 describes three FP utility programs, giving access to the functions of PRIMO.

Chapter 7 describes the information printed from PRIMO to the operator, and the actions the operator may take.

Chapter 8 describes how to generate and install a PRIMO system.

2. SOME FILE ROUTER CONCEPTS 2.

This chapter explains that part of the File Router concepts that is used in connection with PRIMO.

2.1 Transports 2.1

When data are to be moved from one file to another, a transport must be defined. Associated with a transport is a transportdescription in which information about the sending and receiving file is stored together with information about the state of the transport.

Each transport is assigned a unique identification, the transportnumber. This integer is returned from PRIMO, when a transportdefinition is accepted. The transportnumber may later be used in requests to PRIMO for information about the current state of the transport.

A transportname, defined by the user, may be associated with a transport. It cannot be used as identification in questions to PRIMO because it may not be unique. It is, however, returned from PRIMO in answers to requests based on the transportnumber and may be used as an identification for the user.

A username, defined by the user. It can be used as an identification of input to PRIMO, because the username is displayed in messages to the operator concerning the transport. For some output devices it may also be used as an identification of output, because the username is printed on output.

2.2 Devices 2.2

Definitions of a transport must include a description of the sender file and the receiver file, between which data should be moved. In PRIMO, this is done by means of catalog entries, see ref. [4].

The catalog entries must have one of the following formats:

1) An area entry.
   The name of the catalog entry corresponds to an existing backing storage area.
   Example:

```
textfile = set 20 disc2
```

2) A file descriptor describing an existing backing storage area.
   Example:

```
t14 = set bs textfile d.0 0 14
```

   t14 describes a file starting in segment 14 of the backing storage area textfile.

3) A file descriptor referring to an existing external process.
   In this case word 7 and word 8 in the catalog entry tail should be 0.
   Example:

```
lp = set lp printer
```

4) A file descriptor describing a device in RC-NET.
   In this case word 2-5 (document name) in the name of the device at the device host, and word 7 and word 8 are the hostno and hostid respectively of the device host.
   Example:

```
pchremote = set tpe punch 0 17 5033.
```

5) A file descriptor describing an IBM 3270 printer.
   The above mentioned paragraphs (3 and 4) apply to IBM 3270 printers with the following extensions:

   1) The name referred to must be that of the output-link.

   2) The catalog entry 'kind' must be 14, (i.e. lp).

   3) The name must contain the substring 'out', (i.e. 'gout').

4) Word 10 of the catalog entry tail describes the physical address of the referred printer:
<line> shift 16 + <CU> shift 8 + device.

5) Word 10 of the catalog entry tail (line, CU, device) must be different from zero.

The mode field in the file descriptor, see ref. [4], is used when reading from the sender and when writing to the receiver. The mode field specifies a certain hardware or software mode. A detailed description of input/output modes is found in ref. [3].

The kind of the device is stated in the kind field of the file descriptor. In the present version of PRIMO the following kinds of devices are handled:

backing storage	4
typewriter	8
tape reader	10
tape punch	12
printer	14
card reader	16
IBM 3270 printer	14

The catalog entries are looked up on a base interval corresponding to the current catalog base of the process requesting the transport. PRIMO accepts the catalog entries to be as well temporary as permanent. However, if you are running under BOSS or SOS, temporary files should not be used, because they will be removed by the operating system when the job terminates (successfully or in error).

2.3 Criteria 2.3

A transport may specify that certain criteria should be fulfilled by the devices before the transport proceeds.

The criteria are specified by stating the name of a queue and the name of a group. The group is a collection of queues. As an example, a group may exist with the name 'paper'. Within this group a number of queues with names such as 'a4', 'a4cross', 'monitor' etc. may exist. The same queue names may exist within another group e.g. 'control tape', this time referring to the tape that should be mounted in the printer.

The File Router itself does not associate any specific meaning with the criteria stated in a transport. The operator may, however, specify for a certain device that only transports with certain criteria associated should be allowed to proceed. As an example he could specify that only transports which are associated with the queue 'a4' within the group 'paper' should be selected for output to a certain printer.

If the installation treats 'a4' as the standard papertype it may be specified that transports which select queue 'a4' from group 'paper' and transports which have no specification for the group 'paper' at all should be allowed to proceed.

PRIMO supports these facilities in the following restricted way:

1) PRIMO only accepts criteria concerning the receiving device. Only one criterion may be stated, and this must include the group name and one queue name.

2) The criterion is dummy for all devices except printers.

3) The next transport selected for execution on a printer is:

   a) The oldest transport with the same criterion stated in the transport as the one stated in the previous transport. The execution is started without operator intervention.

   b) If no such transport exists then the oldest transport defined is selected.
   A message is output to the operator and PRIMO awaits a reply before printing is resumed.

4) PRIMO allows operators of devices to interact with normal execution. Apart from the 'start', 'stop' and 'kill' commands which are valid for all devices except 'tw', PRIMO offers commands for directing the strategy at transport selection, transport intervention and transport rerouting.

2.4 Transport States 2.4

After definition of a transport, the user may ask for information about its current state.

A transport may be in one of the following states:

waiting, which means that it is in a queue waiting for access to the devices it needs.

held, meaning that it has been granted access to the devices, but waits for some operator action to occur, before it can proceed.

executing, meaning that data are being transferred.

completed, meaning that the transport terminated normally, e.g. because the input file is exhausted.

aborted, meaning that the transport terminated due to some error on one of the devices. In this case additional information on the cause is supplied.

killed by operator, meaning that the operator of one of the devices that are involved in the transport has asked PRIMO to terminate the transport before normal completion.

killed by application, meaning that an application has asked PRIMO to terminate the transport before normal completion.

3. PRIMO INTERFACE SEMANTICS 3.

When an application program wants PRIMO to execute some operation, it sends a data area to PRIMO, where the operation is described. This data area is called the output area.

PRIMO inspects the data area, executes the operation and returns a reply to the application program. This reply is also described in a data area: the input area.

For each operation, PRIMO expects a number of call parameters, and the result of an operation is reported in a number of return parameters.

Most of the fundamental concepts, on which the interaction with PRIMO is based, may be found in chapter 2, as they reflect some general terms used in connection with a transport.

There are in the reply from PRIMO some parameters, which require a more detailed description.

3.1 Reply Information 3.1

Is used in all replies from PRIMO to tell the application program the result of the operation asked for. It is specified by the fields: reply code, sender error and receiver error.

reply code = 0: operation accepted
= 2: transport unknown
= 3: missing resources
= 5: sender troubles
= 6: receiver troubles

sender error: only occurs together with reply code = 5, and has two subparameters, cause and status.

cause = 1: entry troubles
cause = 2: device troubles
status = 0: (unused for the moment)

receiver error: the parameter occurs together with reply code = 6, and has the same meaning as sender error.

3.2 Transport State Information 3.2

Is specified by a number of parameters, transport state, termination position, error cause and error status.

transport state
= 2 waiting, i.e. not yet started or suspended
= 3 executing
= 4 held, i.e. started but waiting for some operator intervention
= 5 completed
= 6 aborted
= 7 killed by operator
= 8 killed by application

error cause
only together with transport state = 6.
= 1 sender device troubles
= 2 receiver device troubles
= 3 operator device troubles

character position
only together with transport state = 5 or transport state = 6. The final character position in the backing storage area, i.e. a pointer to the first character after the last read or written character.

status
only together with error cause. Status is interpreted as described in ref. [1].

3.3 PRIMO Operations 3.3

Below is described for each PRIMO command its function and which parameters are necessary. Some optional parameters are defined and listed too.

3.3.1 Define Transport 3.3.1

Defines a PRIMO transport which causes one file to be copied into another file.

PRIMO will hold the corresponding transport description until a release transport command is received, or an installation dependent time quantum has run out after completion of the transport.

The following parameters are allowed in output area:

transport name
user name
sender name
receiver name

Optional: one queue specification connected to receiver device.

In input area at least the following parameters will occur:

reply code (only value 0, 3, 5 or 6)
transport number
transport name

Optional: sender error
receiver error

In the present version PRIMO handles transports between the kinds of devices stated in the following subsections.

The operator is asked to take action when a hard error occurs (paper out, printer cannot be reserved etc.), or when the criteria of the next selected transport are different from the criteria of the last executed transport.

At any time the printer coroutines active period (that is as long as it has transports to execute) may the operator enter 'select' commands specifying the next transport to be selected. If the coroutine is in a state where the transport selected has not yet been activated, the 'select' command will cause the coroutine to reenter its 'select loop'.

The 'suspend' command offers the possibility of suspending the current transport while printing a more important one. The priority of the transport is not affected by the 'suspend' command.

If the transport is in a state where the transport selected has not yet been activated, the operator may use the 'route' command to direct the transport to another printer device - this is also true for a transport that has been suspended, in which case the transport is regarded as suspended by the receiving printer device (i.e. information about the position from where to resume the transport survives the rerouting).

Hard errors on the backing storage area will always make PRIMO abort the transport (error cause = sender device troubles).

The backing storage area contents are expected to obey the following format:

1) A sequence of ISO characters not containing an EM character (the data file).

2) An EM character.

The output from PRIMO on the printer will appear like:

1) Leading information (headed by start information)
2) Form feed (FF)
3) The data file
4) Form feed (FF)
5) Completion cause
6) Trailing information
7) Form feed (FF)

Start information may be:
empty (no operator intervention),
'operator start'
'operator repeat'
'operator skip'
'operator restart'

Completion cause may be:
empty (normal termination),
'operator kill'
'application kill'
'receiver device status 8. <octal status>' (printer trouble),
'sender device status 8. <octal status>' (disc area trouble) or
'operator device status 8. <octal status>' (operator console trouble).

It should be noted that it is possible to switch off the leading and trailing information (including the form feeds), by an installation parameter, and by the operator command 'triang'.

Example:

```
Leading information
OPERATOR START
***************
 *************
  ***********
   *********
    *******
     *****
      ***
       *
PRIMO       :82.08.01. 12.00
CONTENTS OF:TESTDATA
TRANSPORT   :
USER        :FB

Trailing information
       *
      ***
     *****
    *******
   *********
  ***********
 *************
***************
PRIMO       :82.08.01 12.01
CONTENTS OF:TESTDATA
TRANSPORT   :
USER        :FB
```

3.3.1.2 Backing Storage Area to Punch 3.3.1.2

The operator of the punch is always asked to take actions before a transport is executed.

The backing storage area contents are expected to obey the following format:

1) A sequence of ISO characters not containing an EM character (the data file).

2) An EM character.

The output from PRIMO on the punch will appear like:

1) 90 NUL characters.

2) The data file (with a character coding corresponding to the mode defined in the entry of the transport).

3) 90 NUL characters.

If a harderror (e.g. paper out) occurs on the punch, the output is split on several tapes.

3.3.1.3 Paper Tape Reader to Backing Storage Area 3.3.1.3

The statements given in subsection 3.3.1.2 about the operator of the punch are also valid for the operator of the paper tape reader.

The input from the reader must be a number of paper tapes, with a character coding corresponding to the mode defined in the entry describing the input device (e.g. ISO code odd parity = 0, ISO code even parity = 2, ISO code no parity = 4, flexowriter code = 6).

The output from PRIMO will appear like:

1) A sequence of data blocks received from the reader.

2) Each segment consists of an integral number of input blocks, and is filled in with NUL-character.

3) The last segment is filled in with EM characters.

3.3.1.4 Card Reader to Backing Storage Area 3.3.1.4

The statements given in subsection 3.3.1.2 about the operator of the punch are also valid for the operator of the card reader.

The input from the card reader must be a number of cards, with a character coding corresponding to the mode defined in the entry describing the input device (e.g. punched binary = 0, punched

decimal with conversion = 10, mark sense binary = 64, mark sense decimal with conversion = 74, basic cards = 256).

The output from PRIMO will appear like:

1) A sequence of card images.

2) Each segment constists of an integral number of input blocks, and is filled in with NUL-characters.

3) The last segment is filled in with EM characters.

3.3.1.5 Teletype to Backing Storage Area 3.3.1.5

This transport is intended for copying of paper tapes from a teletype reader to a backing storage area.

The teletype itself is defined as operator device for the teletype.

The input should be a paper tape with a character coding corresponding to the mode defined in the entry describing the input device (e.g. ISO code even parity = 2, ISO code no parity = 4).

The output from PRIMO will appear like:

1) A sequence of data blocks received from the teletype.

2) Each segment consists of an integral number of input blocks, and is filled in with NUL-characters.

2) The last segment is filled in with EM characters.

3.3.1.6 Backing Storage to IBM 3270 Printer 3.3.1.6

This kind of transport is intended for copying of backing storage area to printers connected to the IBM 3270 terminal handler.

The contents of the backing storage are expected to obey the following format:

1) A sequence of ISO characters not containing an EM character (the data file).

2) An EM character.

Output from PRIMO on the printer will appear as a number of output blocks containing an integral number of lines.

Each block is headed by a 'block header' and terminated by a 'block trailer'.

The block header contains:

+0: CU, device, ESC (27)
+2: Write Code (53), WCC (8), USM (31)! CR (13)
+4: (CR (13)) 0/3, (char) 0/3

If the block is the last block the USM is replaced by CR-character to release the printer reservation within the CU.

The block trailer contains:

n-2: (char) 0/3, (EM (25)) 0/1, (NULL (0)) 0/1
n: (EM (25)) 0/1, (EXT (3)), (NULL (0)) 1/2

CR (13) is used as fill in the header when positions in a partial word have been printed in a previous block trailer.

If the transport is aborted, PRIMO will print, if possible, the cause for abortion on the printer:

'*** killed by operator'
'*** killed by application'
'*** operator device trouble'

3.3.2 Get State of Transport 3.3.2

Is used to ask for the current state of a transport, previously defined by 'define transport'.

The output area holds the following parameter:

transport number

In input area at least the following information will occur:

reply code (only value 0 or 2)
transport number
transport name
subtransport state

optional: error cause
error status

3.3.3 Wait and Get State of Transport 3.3.3

Is used to wait for completion of a transport previously defined by 'define transport'. The transport can be either successfully terminated, or terminated by some hard error during the transport. The formats for output and input area are as for 'get state of transport'. Subtransport state in the input area tells if the termination was successful or not.

3.3.4 Release Transport 3.3.4

Releases a transport description as soon as the given transport is terminated.

Output area holds the following parameter:

transport number

Input area holds at least the following parameters:

reply code (only value 0 or 2)

3.3.5 Kill Transport

If the transport is already executed nothing is done. Else actions are initiated, which will terminate the transport, when communication in progress with a device or an operator is completed.

4. PRIMO INTERFACE FORMATS 4.

An application interfaces PRIMO by means of the send message/wait answer procedures of the monitor, see ref. [2].

The message used is a combined output/input message. The operation and the call parameters are specified in the output area, and the result of the operation is returned in the input area.

4.1 Message 4.1

A message with operation code = 7 is sent to the process named 'primo'. The message has the following format:

+0	7 < 12
+2	first output area address
+4	last output area address
+6	first input area address
+8	last input area address

Output area and input area need not be different areas.

4.2 Answer 4.2

Status is normally 0, all parameter errors are signalled in the input data area.

status = stopped
Occurs when the sending process was stopped during PRIMO inspection of one of data areas. In this case the function is not executed.

no of bytes = length of input area

no of characters = length of input area

4.3 Formats of Output Area and Input Area 4.3

4.3.1 Define Transport 4.3.1

Output Area

wordno \ bitno	0 ...	11..	15..	19..	23
1	2	0	0	0	
2	1	1	0	4	
3					
4					
5	TRANSPORT NAME				
6					
7	2	1	0	4	
8					
9	USER NAME				
10					
11					
12	1000	1	0	0	
13	1	2	0	0	
14	2	3	0	4	
15					
16					
17	NAME of sender				
18					
19	2	2	0	0	
20	2	3	0	4	
21					
22					
23	NAME of receiver				
24					

With one queue criterion defined:

25	3	3	0	0	
26	1	4	0	4	
27					
28					
29	GROUP NAME				
30					
31	3	4	0	4	
32					
33					
34	QUEUE NAME				
35					

3	0	0	0
1	1	0	4
TRANSPORT NAME			
2	1	0	4
USER NAME			
3	1	0	1
TRANSPORT NUMBER			
4	1	0	0
1	2	0	1
REPLY CODE			

If reply code is 5 or 6 the input area is expanded with the following fields:

reply code = 5

3	2	0	2
SENDER CAUSE			
0			

reply code = 6

4	2	0	2
RECEIVER CAUSE			
0			

4.3.2 Get State of Transport 4.3.2

Output Area

1	4	0	0	0
2	3	1	0	1
3	TRANSPORT NUMBER			

Input Area

5	0	0	0
4	1	0	0
1	2	0	1
REPLY CODE			
1	1	0	4
TRANSPORT NAME			
3	1	0	1
TRANSPORT NUMBER			
1000	1	0	0
3	2	0	0
4	3	0	1
TRANSPORT STATE			

If transport state specifies that the transport is completed or aborted, the following field is added:

7	3	0	2
CHARACTER POSITION			

If transport state specifies error termination one more field is added:

6	3	0	2
ERROR CAUSE			
STATUS			

4.3.3 Wait and Get State of Transport

4.3.3

Output Area

1	6	0	0	0
2	3	1	0	1
3	TRANSPORT NUMBER			

Input Area

7	0	0	0
4	1	0	0
1	2	0	1
REPLY CODE			
1	1	0	4
TRANSPORT NAME			
3	1	0	1
TRANSPORT NUMBER			
1000	1	0	0
3	2	0	0
4	3	0	1
TRANSPORT STATE			

If transport state specifies that the transport is completed or aborted, the following field is added:

7	3	0	2
CHARACTER POSITION			

If transport state specifies error termination one more field is added:

6	3	0	2
ERROR CAUSE			
STATUS			

4.3.4 Release Description 4.3.4

Output Area

1	10	0	0	0
2	3	1	0	1
3	TRANSPORT NUMBER			

Input Area

11	0	0	0
4	1	0	0
1	2	0	1
REPLY CODE			

4.3.5 Kill Transport

Output Area

1	10	0	0	0
2	3	1	0	1
3	TRANSPORT NUMBER			

Input Area

11	0	0	0
4	1	0	0
1	2	0	1
REPLY CODE			

5. EXTERNAL ALGOL PROCEDURE TRANSFER 5.

General Description

This ALGOL procedure is used to access PRIMO in a more convenient way than possible, when directly using the specifications for messages to the PRIMO process. The cost for this is that only a limited set of parameters is allowed. The parameters allowed are the most usual ones.

With this procedure it is possible to execute the following PRIMO functions:

1) define a transport
2) ask for current state of a defined transport
3) wait for termination of a defined transport
4) release transport
5) kill transport

If the process running the application program is stopped during communication with PRIMO, the procedure repeats the function, until it is executed.

Below is given the exact formats for call of the procedure. In general the array carr is designed for extension with some optional parameters. Optional parameters are given the value -1 before call, signalling that these parameters should not be transferred to PRIMO. In the same manner, by return from the procedure, parameters having value -1 in rarr were not present in the answer from PRIMO.

5.1 Call 5.1

call: transfer (action, carr, cleng, rarr, rleng)

transfer: (return value, integer). Specifies the result of the execution of the procedure.

= 0: function executed

<> 0: function not executed due to some error in parameters or to missing resources. The most common is that the PRIMO module is not present, that the process calling the procedure has no message buffer resources left or some error in the parameters to the procedure.

2: PRIMO message rejected. Can only occur when another user already has sent a 'wait and get state' function for the same transport.

3: PRIMO MESSAGE UNINTELLIGIBLE: Should not occur.

4: PRIMO malfunction.

5: PRIMO does not exist.

6: Message buffer claim exceeded. The procedure uses one message buffer for communication with PRIMO.

7: Parameter action has an illegal value.

8: Parameter cleng or rleng is not large enough to allow the corresponding array to hold the function defined by action.

9: Only possible for action = 2. Tells in this case that the type of the criterion, found in the data area, is not legal.

action: (call value, integer). Identifies the PRIMO function to be executed. The following functions are defined (Detailed description is found in chapter 3):

= 2: define transport
= 4: get state of transport
= 6: wait and get state of transport
= 8: release transport
= 10: kill transport

car: (call value, integer array). Action identifies the PRIMO function to be executed, and the corresponding parameters are handed over to the procedure in the array carr. Detailed format is found below.

cleng: (call value, integer). Length of array carr.

rarr: (return value, integer array). Action, carr has defined the PRIMO function to be executed, rarr contains at return from the procedure the answer parameters from the call of PRIMO. Detailed format is found below.

rleng: (call value, integer). Length of array rarr.

5.1.1 Define Transport (action = 2) 5.1.1

Format of carr.

With no queue specification:

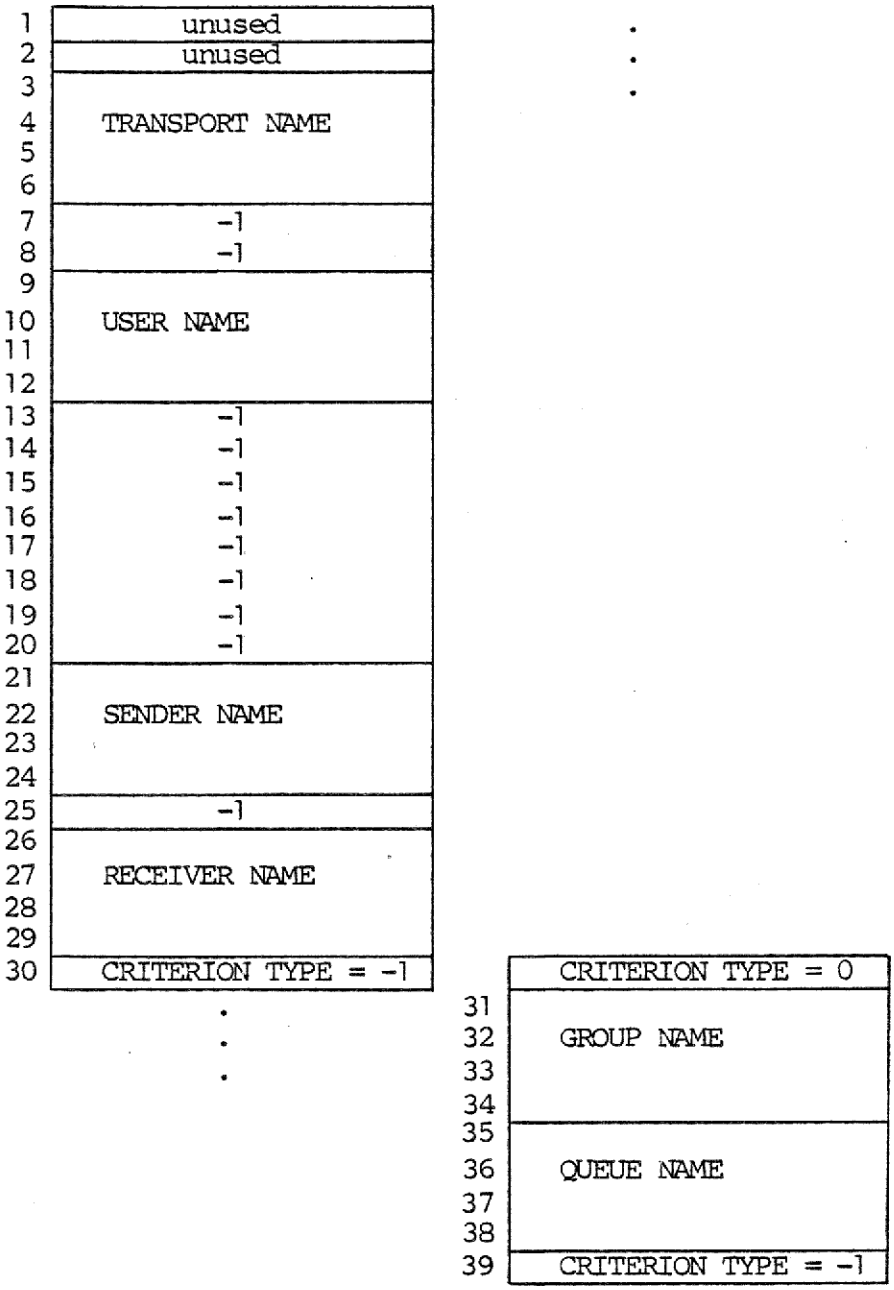
```
integer array carr (1:30)
```

With one queue specification:

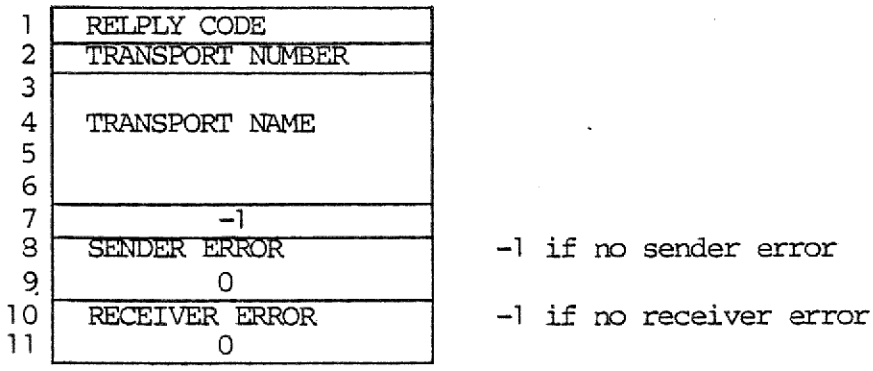
```
integer array carr (1:39)
```

To signal to 'transfer' the end of queue specification a pseudo specification of one word with criterion type = -1 has to occur in carr.

With no criterion



Format of rarr, integer array rarr (1:11)



5.1.2 Get State and Wait and Get State of Transport 5.1.2

integer array carr (1:9)

1	unsused
2	TRANSPORT NUMBER
3	-1
4	-1
5	-1
6	-1
7	-1
8	-1
9	-1

Answer get state and wait and get state of transport

integer array rarr (1:26)

1	RELPLY CODE	
2	TRANSPORT NUMBER	
3		
4	TRANSPORT NAME	
5		
6		
7	-1	
8	-1	
9	-1	
10	-1	
11	-1	
12	-1	
13	-1	
14	-1	
15	-1	
16	-1	
17	-1	
18	-1	
19	-1	
20	1	
21	TRANSPORT STATE	
22	CHARACTER PSOITION	or -1
23		
24	-1	
25	ERROR CAUSE	or -1
26	ERROR STATUS	or -1

5.1.3 RELEASE Transport and Kill Transport

integer array car (1:7)

1	unused
2	TRANSPORT NUMBER
3	-1
4	-1
5	-1
6	-1
7	-1

Answer release transport and answer kill transport

integer array rarr (1:16)

1	REPLY CODE
2	TRANSPORT NUMBER
3	-1
4	-1
5	-1
6	-1

5.2 Resources

The procedure uses the following resources:

1) One message buffer for communication with PRIMO.

2) Approx. 600 halfwords in the stack for variables.

6. FP UTILITY PROGRAMS FILEXFER, FILEENQ AND SAVETRANS 6.

This chapter describes two FP utility programs, filexfer and fileenq, which may be used to define transports, to get information about the current state of a transport and to wait for the completion of a transport.

6.1 Filexfer 6.1

Defines a File Router transport, which causes one file to be copied into another file.

Examples:

The FP command:

```
fielexfer myfile lp
```

will cause the file 'myfile' to be moved to the file 'lp' which normally will be a filedescriptor describing the standard printer.

The FP command:

```
filexfer myfile lp queue.paper.a4
```

will cause the same file to be moved, but the operator will be asked to mount the papertype 'a4' in the printer before the transport proceeds.

The FP command:

```
t = filexfer tdstat lpt queue.forms.tdstd release.no
```

will cause the file 'tdstat' to be moved to the file 'lpt', which could be a remote printer. Before the transport is initiated, the operator will be asked to prepare the printer with forms of the type 'tdstd'. The meaning of this name depends on the installation. The parameter release.no specifies that the transport description shall not be released when the transport is finished.

The left side causes a catalog entry named 't' to be created. This entry may later be used as parameter to the program fileenq to ask for information about the state of the transport or to wait for the transport to finish.

The FP commands:

```
workarea = set 100

filexfer tre workarea wait.yes
if ok.yes
filexfer workarea lp name.tapecopy
```

will copy the file described by 'tre' (normally a reader) to the file described by 'lp' (normally a printer). The copying is performed in two steps: A transport from 'tre' to the bs-area 'workarea' followed by a transport from 'workarea' to 'lp'. In the first call of filexfer the wait.yes parameter is specified. Therefore the definition of the last transport is delayed, until the first one is completed.

Call:

$$\left\{\text{<result name>=}\right\}_{0}^{1}\ \text{filexfer <infile> <outfile>}\ \left\{\text{<params>}\right\}_{0}^{\infty}$$

$$\text{<params>::=}\ \left\{\begin{array}{l}\text{name.<transport name>}\\ \text{user.<user name>}\\ \text{queue.<group name>.<queue name>}\\ \text{wait.}\left\{\begin{array}{l}\text{yes}\\ \text{no}\end{array}\right\}\\ \text{release.}\left\{\begin{array}{l}\text{yes}\\ \text{no}\end{array}\right\}\\ \text{verify.}\left\{\begin{array}{l}\text{yes}\\ \text{no}\end{array}\right\}\end{array}\right\}$$

<infile>::= <outfile>::= <transport name>::= <user name>
<group name::= <queue name>::= <text>

Function:

A transport will be created. It will specify a movement of data from the file specified by <infile> to the file specified by <outfile>. In the present version of PRIMO the following combinations of input files and output files are allowed:

<infile>	<outfile>
bs-area	printer
bs-area	punch

paper tape reader	bs-area
card reader	bs-area
typewriter	bs-area
bs-area	IBM 3270 printer

If <result name> is present, a temporary catalog entry will be created with the name specified. It will contain a description of the transport in a way such that it can be used later as parameter to the program fileenq.

The name-parameter assigns a transport name to the transport. The name may be used for identification purposes. The name-parameter may appear only once in the parameter list. If no name-parameter is present, a null-name will be used.

The user-name assigns the name of the user to the transport. If no user-name parameter is present the name of the defining process will be used.

The queue-parameter is used to specify certain properties, that should be fulfilled by the receiving device before the transport can be started. The parameter causes the transport to enter a queue specified by <queue name>. The queue is part of a group of queues specified by <group name> and the interpretation of the parameters depend on the installation. Typical values for <group name> could be 'paper', 'forms', 'tape' etc. while typical names for <queue name> could be 'a4', 'monitor', 'a4double', 'invoice' etc.

The wait-parameter specifies, whether filexfer should wait for the transport to terminate. Standard setting is wait.no.

The release-parameter specifies whether the transport description should be released as soon as it terminates, whether this occurs in error or not. Standard setting is release.yes.

The verify-parameter specifies, whether the identification of the transport (in addition to eventually saving it in <result name>) should be output on current output. The transport identification is an integer. The identification output may be used later as parameter to the program fileenq.

Error Messages:

***filexfer param: <param>

An illegal parameter has been detected. The parameter is shown. Filexfer proceeds to the next parameter.

***filexfer sendername missing

The input device has not been specified.

***filexfer receivername missing

The input device has not been specified.

***filexfer transport name double defined: <param>

The name-parameter occurs more than once.
The parameter is shown and skipped.

***filexfer user name double defined: <param>

The user-parameter occurs more than once. The parameter is shown and skipped.

***filexfer no room for queue specification: <param>

A queue-parameter is met for which there is no room to store the <group name> and the <queue name>. The parameter is shown and skipped.

***filexfer create <result name> <result>

An error has occurred during the creation of the catalog entry with name <result name>. <result> may be either:

entry already exists
claims exceeded
catalog base illegal

or

result: <value>

where <value> is the result of the monitor procedure create entry.

The above mentioned error messages all appear during the scan of the parameterlist. If any error messages have been output,

filexfer will terminate as soon as the parameter list is exhausted. No transport will be defined.

***filexfer primo communication error: <cause>

A severe error has occurred in the communication with the File Router module PRIMO. The most common reasons will be that PRIMO is not present or that the process in which filexfer is called has no message buffer resources left. <cause> may be either

rejected
unintelligible
malfunction
primo does not exist
message buffer claim exceeded

or

unexpected result: <value>

the latter occuring if a result, not prepared for, is received.

***filexfer primo reply error: <reply>

PRIMO has rejected the creation of the transport. <reply> indicates the reason and may be either

missing resources
sender entry troubles
sender device troubles
receiver entry troubles
receiver device troubles

or

unexpected reply code: <value>

the latter again occurring if a reply, not prepared for, is received.

The term entry refers to the catalog entry given as input- or output parameter to filexfer. The term device refers to the device (printer etc.) described in the entry.

Setting of the ok-bit:

If any error message has been output, the ok-bit is set to false. Otherwise the ok-bit is set to true.

6.2 Fileenq

This program is used to ask for the current state of a transport, previously defined by the program filexfer. The termination of the transport may be waited for.

Examples:

The FP command:

```
fileenq t
```

will output the current state of the transport, whose identification is given in the catalog entry 't'. This entry should have appeared as left side parameter in a previous call of the program filexfer.

The FP command:

```
fileenq 1448 wait.yes details.yes
```

will wait for the termination of the transport whose identification is '1448'. When the transport has terminated, its status (terminated ok or in error) will be output. If an error has occurred, the device status will be shown.

Call:

fileenq <identification> $\left\{ \text{<params>} \right\}_{0}^{\infty}$

<params>::= $\left\{ \begin{array}{l} \text{wait.} \left\{ \begin{array}{l} \text{yes} \\ \text{no} \end{array} \right\} \\ \text{release.} \left\{ \begin{array}{l} \text{yes} \\ \text{no} \end{array} \right\} \\ \text{kill.} \left\{ \begin{array}{l} \text{yes} \\ \text{no} \end{array} \right\} \\ \text{details.} \left\{ \begin{array}{l} \text{yes} \\ \text{no} \end{array} \right\} \end{array} \right\}$

<identification>::= { <text> | <integer> }

Function:
The transport is identified by the parameter <identification>. This may be either an <integer>, in which case it should be the number output by a previous call of filexfer with the verify.yes parameter, or it may be a <text> in which case it should be the name of a catalog entry previously created by appearing as left side parameter in a call of filexfer.

The wait-parameter specifies, whether fileenq should wait for the transport to terminate, before the state is printed. The state after termination could be either terminated correctly or terminated in error. Standard setting is wait.no.

The release-parameter specifies whether the transport description should be released or not. If release.yes is specified, the transport description is always released. If release.no is specified, the transport description is not released. If the release-parameter is not present, the transport description is released, provided the state indicates terminated without error.

If the transport description is released, and <identification> is of <text>-type, the corresponding catalog entry will be removed.

The kill-parameter specifies whether the transport should be killed or not. If kill.yes is specified, the transport is killed. Otherwise the transport is not killed.

The details-parameter specifies the amount of information, concerning the state of the transport that should be output. See below.

Error Messages:
***fileenq call
Left side in call of fileenq. The program terminates.

***fileenq param: <param>
An illegal parameter has been detected. The parameter is shown. Fileenq proceeds to the next parameter.

***fileenq lookup <entry> result: <cause>
An error has occurred during the attempt to find the catalog entry, specified to hold the identification of the transport, <cause> may be either

entry does not exist

which will be the most common, or the value returned by the monitor function 'lookup entry'.

***fileenq transport identification missing
The <identification> parameter is missing.

The above mentioned error messages all appear during the scan of the parameter list. If any error has been output, fileenq terminates as soon as the parameter list is exhausted. No state-information will be output.

***fileenq primo communication error: <cause>
A severe error has occurred during the communication with the File Router module PRIMO. The most common reason will be that PRIMO module is not present or that the process in which fileenq is called has no message buffer resources left. <cause> may be either:

rejected
unintelligible
malfunction
primo does not exist
message buffer claim exceeded

or

unexpected result: <value>

the latter occurring if a result, not prepared for, is received.

***filexfer primo reply error: <reply>

PRIMO has indicated an error in the attempt to find the transport specified. <reply> may be

transport unknown
missing resources

or

unexpected reply code: <value>

the latter again occurring if a reply, not prepared for, is received.

Format of state information:

If the reply from the File Router indicates that some state information about the transport specified has been delivered, the information is printed in the following way:

```
transport name: <name>
         state: <state information>
        status: <device status>
 char position: <position>
```

Transport name is printed only if the details.yes parameter has been specified. The status information is printed only if the details.yes parameter has been specified and the state of the transport is aborted. The char position is printed only if the details.yes parameter has been specified and the state of the transport is completed or aborted.

<name> is the name of the transport as defined by the name-parameter to filexfer. If this parameter was not present, the name is the null string.

<state information> may be either

waiting
executing
held
completed

aborted <abort cause>
killed by operator
killed by application
or
an integer giving the state.

<abort cause> will be either

caused by sender device
caused by receiver device
caused by operator device
or
an integer giving the cause.

If the state of the transport is aborted, and the details.yes parameter was present, the device status causing the transport to be aborted is shown. <device status> is a list of mnemonic names of the bits set in the status word.

Setting of ok-bit:
If any error-message has been output by fileenq, the ok-bit is set to false. Otherwise it is set to true.

6.3 Savetrans 6.3

The savetrans program is used to restart unfinished transports after a PRIMO break down.

Example:

```
savespool=move primospool
; save the contents of primospool before PRIMO is restarted after
; a break down

savetrans savespool
; call savetrans after PRIMO has been started again
```

Call:

$$\left\{\text{<outfile> =}\right\}_{0}^{1} \text{ savetrans <inputfile> } \left\{\text{<list.<boolean>}\right\}_{0}^{1}$$

<outfile> ; if specified savetrans will make its output on this file else on 'current out'.

<infile> ; bs-area containing a copy of the area primospool.

list.<boolean> ; <boolean> ::= yes!no

Function:
Savetrans restarts all unfinished transports to PRIMO after a PRIMO break down. The transport information is extracted from the bs-area 'primospool' used by PRIMO to maintain its transports description during run. It is therefore essential that the contents of primospool be moved to another file, (<inputfile>) before PRIMO is started again, in order to run the program savetrans.

The parameter 'list.yes' causes savetrans to produce a listing, possibly on <outfile>, concerning each restarted transport and its parameters.

Error Messages:
***savetrans <explanation>

Explanation may be:
create area not possible <outfile>
impossible to create entry <outfile>
connect area not possible <outfile>
impossible to create area process <outfile>
missing resources in primo
PRIMO has not the necessary resources to execute the transport
sender troubles
problems with sender device; does not exist etc.
receiver troubles
problems with receiver device
rejected,
unintelligible,
primo malfunction and
primo does not exist
fatal error in communication with PRIMO, causes savetrans

to terminate. Most common reason will be that the process PRIMO does not exist.

bufferclaim exceeded

missing buffers

illegal action ???

illegal cleng on rleng ???, and

criterion type illegal

should only occur in connection with incompatible versions of PRIMO and savetrans and maybe in connection with harderror on PRIMO's spoolarea 'primospool'

fatal error

comes in connection with errors concerning PRIMO communication e.g. rejected and primo does not exist

set catalog base, illegal bases

the entry bases of the area specified in the transport may not be used as catalog base by the process. To avoid this always run the process with 'all bases' (-8388607 8388605).

7. OPERATING GUIDE 7.

The operator's tasks in the day to day running comprise the following:

1) the start up
2) interaction with PRIMO during the run
3) closing down
4) how to handle error situations

In the following sections the operating system 's' as defined in ref. [5] is implied. If another operating system is used, commands and messages may appear otherwise.

7.1 The Start-up 7.1

PRIMO runs in a separate process normally called 'primo'. The program to be loaded is called 'bprimo' and function bits 1, 2, 3, 4, 5 must be set. Formulas for computing resource claims are given in appendix D.

Example:

```
ATT S
NEW PRIMO SIZE 15000 buf 14 area 12 base -8388607 8388605
READY

ATT S
FUNCTION 1 2 3 4 5 PROG BPRIMO RUN
READY
MESSAGE PRIMO VERSION: 4.0 830401 830401
MESSAGE PRIMO SIZE 14310
MESSAGE PRIMO AREA 10
MESSAGE PRIMO *** BUF 16
PAUSE PRIMO *** INIT TROUBLES

ATT S
REMOVE
READY
```

```
ATT S
SIZE 15000 AREA 10 buf 16 RUN
MESSAGE PRIMO VERSION: 4.0 830401 830401
MESSAGE PRIMO SIZE 14310
MESSAGE PRIMO STARTED
```

PRIMO is started with too few message buffers. Therefore the run is terminated. The optimal values of coresize, area process and message buffers are displayed. Then the process is removed and started again with a reasonable set of resources.

A complete list of messages, which may appear during start up, is given in appendix B.

7.2 Interaction with PRIMO During the Run 7.2

In this section a general description of the interaction between PRIMO and the operators of devices known by PRIMO is given. Handling of specific types of devices, can be different from the general scheme and is explained in the following sections.

If PRIMO is not able to continue the execution of a transport, because manual intervention on a device is desirable, a request message is displayed to the operator (appendix B).

When the operator has performed the steps necessary for PRIMO to resume the execution of the transport, he answers the request by sending a command to PRIMO.

Example:

```
primo: printer prepare programlist nmj paper.a4

att primo
= start printer
primo: ready
```

PRIMO writes out the transportname, the username and the queue criteria of the next transport selected for the local device with

the process name 'printer'. When the operator has changed the paper to a4, he sends a 'start' command to PRIMO. The command is accepted and the execution of the transport is initiated.

Example:

```
primo: ***punch end document

att primo
= start punch
primo: ready
```

While executing a transport, PRIMO has sensed an 'end document' status on the external process called 'punch'. The operator changes the paper tape and sends a 'start' command to PRIMO. The command is accepted, and the execution is resumed.

Example:

```
primo: ***@ printer rejected
       :
       :
att primo
= start @ printer
primo: ready
```

PRIMO has unsuccessfully tried to reserve a printer. The '@' indicates that the message concerns a remote device, and in this case the device name in the device host is used as identification. The reservation fails probably because another system has reserved the printer. When the device is released, the operator asks PRIMO to resume the transport.

The location of the operator terminal depends on whether the device is local or remote.

7.2.1 Operator Output

Operator messages from PRIMO devices are sent to the terminal that has signed up as an operator to the device (see signup command).

If no operator has signed up or the operator-terminal for some reason is unavailable the operator message is sent to the terminal from which PRIMO was started.

If the operator terminal is not connected to the same device host as the device that sends the operator message - and the device is a remote device - information identifying the device host is added to the operator message.

Examples:

```
'primo: *** printer end document' ; local device
'primo: ***@printer end document' ; remote device on same device
                                    host
'primo: ***@printer end document, host.21.5014 ; remote device on
                                                  device host
                                                  5014
```

7.2.2 Operator Input

7.2.2

A terminal may only sign up to a remote device connected to the same device host as the terminal.

A terminal may only handle devices to which it is signed up as an operator.

However:

The PRIMO startup terminal may handle (incl. signup) any device known to PRIMO.

The PRIMO startup terminal need not be signed up to a device in order to handle it.

A device is identified by <device-id> which may be:

A process name (in RC8000), in this case the device is a local device.

A device name (in the device host) preceded by a '@', in this

case the device is a remote device connected to the same device host as the operator terminal.

A device name (in the device host) preceded by a '@' and followed by '.<hostno>.<hostid>', in this case the device is a remote device connected to the device host identified by <hostno> (e.g. 17, 21, 25) and <hostid> (e.g. 5014, 5095 etc.) (see section 2.2 Devices).

Examples:

```
'start printer'  ; local device
'start @printer' ; remote device on same device host
'start @printer.21.5014' ; remote device on device host 5014
```

Operator commands:

start <device id> $\left\{\text{<number>}\right\}_{0}^{1}$

The execution at the transport is started/resumed. The command may be modified by an optional parameter. The meaning of the modification is device dependant and is explained in the next subsections.
The state at the coroutine must be either 'held' or 'drained'.

restart <device id>
The execution of the transport is restarted from the beginning.
The state of the coroutine must be either 'held' or 'drained'.

Skip <device-id> <number>
An attempt is made to upspace the inputfile the specified number of pages from the current position. If the attempt was successful the execution of the transport is resumed/started. If end of medium was encountered during positioning, a warning message is issued to the operator, who may use the repeat or restart command to resume execution.
The state of the coroutine must be either 'held' or 'drained'.

Repeat <device-id> <number>
The inputfile is backspaced the specified number of pages or un-

til start of file is encountered, whichever occurs first, after which the transport is resumed/started.
The state of the coroutine must be either 'held' or 'drained'.

stop <device id>
The execution of the transport is stopped and the state of the coroutine is changed to 'held'.
The state of the coroutine must be 'active'.

Kill <device-id>
The transport is aborted.
The state at the coroutine must be either 'held' or 'drained'.

suspend <device id>
Information about the position of the current transport is stored for later reactivation and the next transport is selected. The priority of the transport is not changed by the suspend command.
The state of coroutine must be either 'held' or 'drained'.

select <device-id> <queue specification>!<queue position>
queue specification::= <empty>!<groupname>.<queuename>
queue position ::= first! last! next! previous! suspended
Directs the strategy of selecting the next transport. If the coroutine is drained the "select loop" is reentered. If the 'select' cannot be honoured the first (oldest) transport is selected.
The state of the coroutine must be either 'held', 'drained' or 'active'.

drain <device-id>
Directs the coroutine to wait for operator intervention after select of the next transport.
May be entered at any state of the coroutine.

route <device-id> <destination>
<destination>::= name of catalog entry.
Reroutes current transport to a device described by <destination>. If the current transport has been suspended, the information concerning the position will survive at the destination.

After succesful routing the coroutine reenters its select loop. The state of the coroutine must be 'drained'.

triang <device-id> on! off
Tells the coroutine whether to print leading and trailing information.
Defaults are as specified by the option parameter to PRIMO generation: 'prltpage'. At 'signof' the value is restored to default.
May be entered in any state of the coroutine.

Request <device>
<device> ::= <empty>!<device-id>!all
<device> = <empty> - displays requests concerning devices to which the sender is signed up as an operator. If the terminal is the startup terminal requests concerning devices for which there exist no operator are also displayed.

<device> = <device-id> - displays a possible request for <device-id>.

<device> = all - displays requests for all devices.

Signup <device-id> <number>
Assigns the sender (terminal) as an operator to the specified device. The number gives the 'kind' of the device. If the device is waiting for operator intervention the request message is displayed to the (possible new) operator.
May be entered in any state of the coroutine.

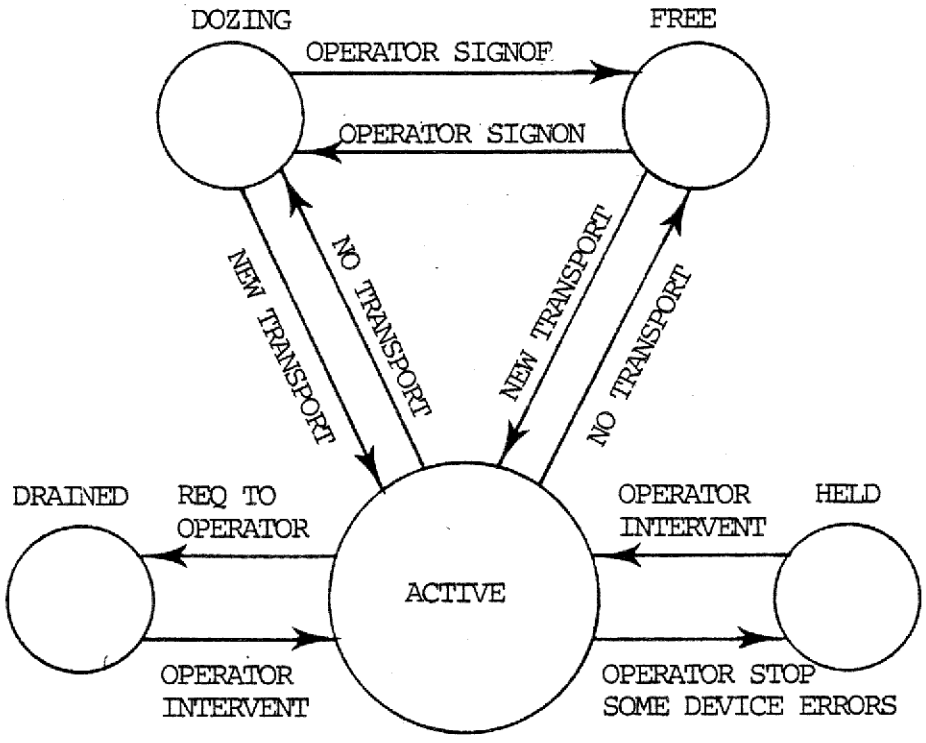
Signoff <device-id>
Cancels the description of the sender as an operator device in the specified coroutine. If the coroutine has no transports it is made available for other devices of the same kind.
May be entered in any state of the coroutine.

Display
Displays information of all coroutines known to PRIMO, provided they are not free, with the following format:

'<device>.<hostno>.<hostid> <procname> <bsname> <oper> <state>'

<device> = device name in device host ; remote device;
<device> = process name in RC8000 ; local device;
<hostno>.<hostid> = identification of device host ; remote
<hostno>.<hostid> = 0.0 ; local device
<procname> = RC8000 process name
<bs_name> = the name of the disc-area used by the transport
<oper> = the name in the device host of the terminal signed up as an operator
<state> = active ; the coroutine is active
<state> = waiting ; waiting for operator intervention
<state> = idle ; the coroutine is signed up but has no transports

The figure below depicts the corelations of states of a coroutine.



Explanation of coroutine states:

free: The coroutine is available for assignment to a device of the coroutines (predefined) kind (initial state of all coroutines). The state may be changed by a 'signup' command (to dozing) or a transport

request for which there is no matching coroutine (to drained or active).

dozing: The coroutine has been assigned to a device, but has no transports at present. The state may be changed by a 'signof' command (to free) or by a transport request (to drained or active).

held: The coroutine has been temporarily stopped by an operator stop command or by some device error requesting operator intervention.

The state may be changed by a 'start', 'skip', 'repeat', 'restart', 'suspend' or 'kill' command (to-active).

drained: The coroutine has been temporarily stopped by some state at device demanding operator intervention or by a previously entered 'drain', 'route' or 'select' command.

active: A transport is being executed by the coroutine. The state may be changed by a 'stop' command, by some device error demanding operator intervention (to held) or by a 'drain-request' (operator 'select' operator 'drain') or some state of device requesting operator intervention (to drained).

7.2.3 Printers

7.2.3

When PRIMO has completed a printer transport it searches for another transport for the printer with matching queue specifications.

If such a transport is found, execution starts without request for operator intervention. Otherwise the first (oldest) transport is selected for execution and the operator is requested to intervene. In case no transport is waiting for execution the

printer coroutine enters the state of dozing or free (if no operator is assigned to the printer coroutine).

However, situations may arise where the above described procedure does not satisfy the demands for effective and flexible device utilization. Below a few hints how PRIMO commands may improve printer utilization are given.

The START, RESTART, SKIP, REPEAT, STOP and KILL commands allow the operator to intervene with the execution of transports - with the purpose of inspection, paper adjusting, ribbon change etc.

The SELECT command allows the operator to decide the priority of transports queued-up for a particular printer.

The ROUTE command offers the operator the possibility to direct a transport to another printer.

The SUSPEND command allows the operator to suspend the printing of a file while executing a more important transport.

The DRAIN command gives the operator the opportunity to have the printer stopped in an orderly way - maybe for the purpose of inspection, ribbon change etc.

The TRIANG command allows the operator to select whether he wants the leading and trailing transport information printed for a particular printer.

Example 1:

```
primo: xxx printer 2 end document
= repeat printer2 2
primo: ready
```

The printer stops because it has run out of paper. After paper change the operator issues a 'repeat' command requesting 2 pages to be repeated.

Example 2:

```
= stop printer2
primo: printer2 stopped by operator
= suspend printer2
primo: ready
primo: printer2 prepare fb forms.a4across
= select printer2 urgent.wages
primo: ready
primo: printer2 prepare pers urgent. wages
= start printer2
primo: ready
```

The operator has been told that a very urgent file must be printed immediately. The operator stops the printing by issuing a 'stop' command, thereafter he issues a 'suspend' command to be able to resume the transport at a later time and finally he selects and starts the pressing transport by issuing 'select' and 'start' commands.

7.2.4 Punches 7.2.4

When PRIMO has completed a transport, the oldest of the transports addressed to the punch is selected for execution. The execution is not started immediately, but the transport is set into hold state, and an operator command is waited for.

Hard errors on a punch caused PRIMO to display the status to the operator and set the transport into hold state. The operator may release the transport by sending a 'start', a 'restart' or a 'kill' command to PRIMO.

7.2.5 Paper Tape Readers 7.2.5

When PRIMO has completed a transport, the oldest of the transports addressed to the reader is selected for execution. Before the execution is started, the transport is set into hold state, and an operator command is waited for.

The operator may use either the 'start' or the 'kill' command.

If the paper loaded is not the last part of the input file, a positive number must be specified as optional parameter to the 'start' command.

Example:

```
primo: reader prepare testdata nmj

att primo
= start reader 1
primo: ready
      .
      .
      .
primo: reader end document
att primo
= start reader
primo: ready
```

First PRIMO asks the operator to prepare the input file to the transport called 'testdata'. The operator loads the reader with the first part of the input file and sends a 'start' command to PRIMO with a positive parameter indicating that the input file continues on another paper tape. When the first paper tape has been read, PRIMO writes out a request to the operator asking for more data. The operator loads the reader with the last part of the input file and sends a 'start' command to PRIMO without the optional parameter. Therefore the transport is terminated, when the paper tape has been read.

Hard errors (except 'end document') on the reader causes PRIMO to abort the transport.

7.2.6 Card Readers 7.2.6

When PRIMO has completed a transport, the oldest of the transports addressed to the reader is selected for execution. Before

the execution is started, the transport is set into hold state, and an operator command is waited for.

The operator may use either the 'start' or the 'kill' command.

If the card batch loaded is not the last part of the input file, a positive number must be specified as optional parameter to the 'start' command.

Hard errors (except 'end document') on the card reader causes PRIMO to abort the transport.

7.2.7 Teletypes 7.2.7

No operator is connected to a teletype in the sense described in the previous subsections.

Whenever an input file is to be read from the teletype, the following message appears on the teletype itself:

primo: transmit

Now the teletype operator is supposed to load the paper tape reader of the teletype with a paper tape within a system defined period (typically 30 seconds).

7.2.8 IBM 3270 Printers 7.2.8

When PRIMO has completed a transport, the oldest of the transports selected for the IBM 3270 printer is selected for execution. Execution is started immediately.

As IBM 3270 devices in some ways differ from standard RC-devices, a short survey is given below concerning operator messages, standard actions and possibilities for operator intervention.

7.2.8.1 IBM 3270 Operator Messages

The device name referred to in operator messages is a compound name, based on the catalog entry tail's docname and the physical address of the device.

The substring 'out' in docname, e.g. gout, is substituted in the following way:

'o' is replaced by 130, (48 + <line>)
'u' is replaced by by (48 + <CU>)
't' is replaced by (48 + <device>)

I.e. an operator message concerning a device of the physical address line = 1, CU = 0, device = 5 and connected via the link 'gout4' would contain the device name "g1054".

Operator messages describing states of device only applicable to IBM 3270 printers:

a) Results of connection:
   illegal line, CU, device
   printer unknown
   printer reserved
   no resources at device host
   printer disconnected

b) Answer to output operations from the control unit, i.e. status bytes status 0 and status 1 (see ref. [6], section 3.5):
   printer unavailable
   printer busy
   printer offline
   printer command
   printer status (S0/S1) = hex. <Status 0> <Status 1>

7.2.8.2 IBM 3270 Standard Actions

PRIMO will wait approx. 2 minutes for the completion of an output

operation; that is, from the output message has been sent until a status (status 0, status 1) has been received from the control unit. If the status received is a 'printer offline' status, the waiting-period is extended to approx. 10 minutes.

If no status is received during the waiting-period, the transport is aborted with status: Timer.

Except from 'printer offline' any of the above mentioned messages indicate a state that will cause the current transport to be aborted with cause = receiver device troubles, state = aborted and status = normal answer.

The IBM 3270 coroutine only accepts the 'start', 'stop' and 'kill' commands.

7.3 Closing the System 7.3

In PRIMO no close command exists. Closing down after a normal run is done by simply removing the process as seen below:

```
att s
proc primo remove
```

7.4 How to Handle Error Situations 7.4

During the run the system may break down in one of the following ways:

1) A program error may cause the system to break down, and the following error message will be printed on the terminal from where the system was started:

```
pause primo ***fault
```

2) The system dies without printing a message. Then the process ought to be 'breaked' in order to have the last portion of testoutput generated, written on the testarea:

   att s
   proc primo break

3) A hard error in a work area makes continued running impossible and the system stops after having printed the message:

   pause primo status <status word> area

In all error situations one should, if the system has been trimmed with 'testoutput', move this from the test area PRIMOTEST to a work area, from which the TRACE-program can print it for a further analysis.

The TRACE program is automatically generated by the installation of the system. The program is called as follows:

   trace <testarea>.<segments>

<testarea> is the name of the area, from which the testoutput is to be printed (the work area the testoutput has been moved to, or the test area itself).

<segments> are the maximum number of segments to be analyzed. TRACE always finds the latest generated segments, and counts the number of segments backwards from there. <segments> are automatically cut to the size of the area, if something larger has been specified.

Examples:

1) An s run; testoutput is printed before a restart.

```
att s
proc primo remove new primo run
ready
o lp
trace primotest.10000 (everything is printed)
o c
att s
proc primo remove
-          (a new start-up)
```

2) A BOSS run; the testoutput has been moved to the area TESTCOPY

```
10 o pip
20 trace testcopy.10000 (everything is printed)
30 o c
40 convert pip
50 finis
go
```

8. SYSTEM GENERATION

8.1 Installation

PRIMO may be installed on all models of the RC8000 series computers. Before installation check that

a) the version of your monitor $\geq$ 5.0
b) your ALGOL compiler is ALGOL7 or newer.

If one of these conditions is not fulfilled, the installation will terminate unsuccessfully, or the execution of PRIMO will stop because of runtime errors.

In order to utilize resources in an adequate way it is possible to adapt PRIMO to fit special needs of an installation. Therefore one has to consider the following trim parameters before installation.

"options" — At start-up a message showing the version and the data of PRIMO will be displayed together with this constant. At each trimming this constant should be changed to show the date of the trimming (e.g. 790601). The standard value is 0, indicating that standard trimming is used.

"prcount" — The number of printer coroutines in use at the same time.
PRIMO uses one printer coroutine to serve each printer known to PRIMO. A printer is known from the device is signed up by the operator (chapter 7), or a transport is defined to the printer (whatever happens first), and until the last transport concerning the printer is executed, and the printer is signed off by the operator.
Restrictions: Min. 0.

"prbufsize" The size in halfwords of the buffers used for printer output.
Restrictions: Min. 104, max. 512, even.

"prltpage" The value defines whether printer lists as a default should be headed and terminated by a page identifying the transport.

prltpage = 0 No leading and trailing page are printed.

prltpage <> 0 Leading and trailing page are printed on all printer lists.

"prlinepage" The number of lines per printer page. The value is used in some operator commands concerning printer pages.

If N NL characters are found between two FF characters, PRIMO defines the number of pages in this part of the file to be

N // prlineprpage + 1.

Restrictions: Min. 1.

"pccount" Defines the number of punch coroutines. Consumption of punch coroutines: see the description of "prcount".
Restrictions: Min. 0.

"pcbufsize" The size in halfwords of the buffers used for punch output.
Restrictions: Min. 104, max. 512, even.

"rdcount" Defines the number of paper tape reader coroutines. Consumption of paper tape reader coroutines: see the description of "prcount".
Restrictions: Min. 0.

"rdbufsize" The size is halfwords of the buffers used for paper tape reader input.
Restrictions: Min. 104, max. 512, even.

"twcount" Defines the number of type writer coroutines. Consumption of type writer coroutines: see the description of printer coroutines.
Restrictions: Min. 0.

"twbufsize" The size in halfwords of the buffers used for type writer input.
Restrictions: Min. 104, max. 512, even.

"fprcount" Defines the number of IBM 3270 printer coroutines. Comsumption of IBM 3270 printer coroutines: see the description of IBM 3270 printer coroutines.
Restrictions: Min. 0, max. 49.

"fprbufsize" The size in halfwords of the buffer used for IBM 3270 printer output.
Restrictions: Min. 104, max. 512, even.
(should at least accomodate a full print line inclusive header and trailer information (+10 halfwords)).

"oprcount" The number of operator coroutines. Defines the maximal number of operators having a conversion with PRIMO at the same time.

A conversation is started when an operator after having pressed the attention button addresses PRIMO, and is terminated, when PRIMO has written the reply on the operator command.
Restrictions: Min. 1.

"trsegm" The size in backing storage segments of the transport description area. The area holds

descriptions of all transports known by PRIMO. A transport is known from the transport is defined, and until a certain time defined in the trimming (see "trsaveminut") has passed after completion of the transport. It is possible for an application program to release the transport description earlier, if an explicit release operation is sent to PRIMO.
Restrictions: Min. 0.

"testsegments" The size in backing storage segments of the testoutput area. If this number is zero, no testoutput is generated. Performance is higher if testoutput is suspended, but the possibilities for recovering system errors will be minimal.
Restrictions: Min. 0.

"waitops" The number of pending "wait and get state of transport" operations (see chapter 3).
Restrictions: Min. 0.

"oprdetails" Defines whether details about the termination state of the transports should be displayed to the operator.

oprdetails = 0: no details are displayed.

oprdetails <> 0: details are displayed.

The system trimming is done by means of the file primotrim (see appendix C), which contains a set of standard variables plus commands for generating the trimmed version of the program.

Installation may be done after that the files have been loaded to disc or directly from tape.

a) Installation from tape.

If the system tape is called mtprimo the installation is performed as shown below:

```
primodoc     = set 1 <discname>; default = disc
mipshelp     = set mto mtsw8100 0 2
i mipshelp
i primohelp
xtrim        = edit primotrim

                EDIT COMMANDS

i xtrim
```

b) Installation from diskette.

If the system diskettes are called S18100 and S28100 the installation is performed as shown below:

"fdload S18100.1"
fdload itself will ask for mounting of the continuation volume.

When the files have been loaded, use the fp command:

'i mipshelp'

You may now proceed with paragraph c), installation from backing storage.

c) Installation from backing storage.

```
primodoc     = set 1 discname ; default = disc
i primohelp
xtrim        = edit primotrim

                EDIT COMMAND

i xtrim
```

The process used for installation may run with standard resources except that

a) coresize must be $\geq$ 50000 halfwords, 60000 reasonable,

b) backing storage resources must be available for the files:
   bprimo
   trace
   transfer
   filexfer and
   fileenq
   savetrans
   The files will be installed as permanent files with scope equal to user scope. If the files already exist, the old contents will be overwritten,

c) user scope must equal system scope (- 8388607:8388605). If this is not the case, the scope of the file bprimo must be changed by hand to system scope after installation.

Example:
The installation is done from the tape mtsw8100 and the trim parameters prcount and prbufsize is changed to 8 and 256 respectively.

```
primodoc  = set 1 disc2
mipshelp  = set mto mtsw8100 0 2
i mipshelp
i primohelp

xtrim     = edit primotrim
1./prcount/,r/1/8/
1./prbufsize/,r/128/256/
f
i xtrim
```

A. REFERENCES A.

[1] RCSL No 42-i1263:
ALGOL8, User's Guide, Part 1

[2] RCSL No 31-D679:
RC8000 Monitor, Part 2

[3] RCSL No 31-D478:
RC8000 Monitor, Definition of External Processes, Part 3
(see section A.1)

[4] RCSL No 31-D676:
System 3, Utility Programs, Part 1

[5] RCSL No 31-D643:
Operating System s, Reference Manual

[6] RCSL No 42-i2156:
RC855 IBM 3270 BSC Emulator, Reference Manual

A.1 Additional References A.1

A detailed description of specific kinds of external processes may be found in:

RCSL No 31-D539:
RC8000 Backing Store Area Process

RCSL No 31-D536:
RC8000 Lineprinter Process

RCSL No 31-D534:
RC8000 Papertape Punch Process

RCSL No 31-D535:
RC8000 Papertape Reader Process

RCSL No 31-D537:
RC8000 Punched Card Reader Process

RCSL No 31-D580:
RC8000 Terminal Process

RCSL No 31-D693:
IBM 3270 Terminal Handler
User's Guide

B. SYSTEM MESSAGES AT START-UP B.

Syntax of a start up message:

{message / pause} primo {<sp><sp><sp> / ***} <message text>

List of start up message text	
Release: <release no><date><option date>	Indicating the date of the system.
size <i>	optimal value of core-size.
area <i>	optimal value of area processes.
buf <i>	optimal value of message buffers.
name of area <i>	too few resources for creating work areas.
started	primo is running.
init troubles	resources missing, execution terminated.

B.1 Request Messages to the Operators B.1

Syntax of request message:

primo {***} <device id> <message text> !
primo {***} <device id> <message text>, host.<hostno>.<hostid>

<device id> :: = <process name> |
@ <device name>

If device id is preceded by '***', the message is an error message.

List of message texts

prepare <transport name> <user name> { <queue group>.<queue name> }
Informs the operator about the next transport in progress for one of the devices he is signed up to.

resume <transport name> <user name> <queue group>.<queue name>
Requests the operator to resume a previously suspended transport for one of the devices he is signed up to.

intervention	bit 12
parity error	bit 13
timer	bit 14
data overrun	stopped
block length	word defect
end document	position err.
load point	does not exist
tapemark, att	disconnected
write enable	unintelligent
mode error	rejected
read error	normal
card reject	

A status error has occurred on the device. Details about the status information is given in ref. [1] and ref. [3].

Stopped by operator

The transport is set into hold state, because the operator has sent a stop command.

Transmit

Only used in connection with teletypes. The operator is expected to load the paper tape reader.

illegal line, CU, device
printer unknown
printer reserved
no resources at device host

PRIMO has not been able to connect the specified printer, see ref. [6].
Only used in connection with IBM 3270 printers.

printer disconnected
PRIMO has received timer status on the output link, which indicates that the printer has been disconnected by a release message from another process, see ref. [6].
Only used in connection with IBM 3270 printers.

printer unavailable
printer busy
printer offline
printer command
printer status (S0/S1) = hex. <S0> <S1>

PRIMO has received a status from a control unit different from the normal status (device end), see ref. [6].
Only used in connection with IBM 3270 printers.

B.2 Log Messages to the Operator

Syntax of a log message:

primo: <device id> <message text> !
primo: <device id> <message text>,host.<hostno>.<histid>

<device id>:: = <process name> |
@ <device name>

List of log texts

end transport <transport state>
Displayed to the operator when a transport is terminated (this message appears only when 'oprdetails' is set <> 0 in the installation of PRIMO).

B.3 Answers from PRIMO on Operator Commands

Syntax of answer:

primo: *** <answer text>

If the <answer text> is preceded by '***' the message is an error message.

List of answer texts

ready	The command is accepted.
syntax	Syntax error in the command (error).
command unknown	The command typed is not a PRIMO command (error).
command + param	Too many parameters (error).
command - param	Too few parameters (error).
device unknown	PRIMO does not know the device (error).
state illegal	The command is not legal in the current state of the transport (error).
not allowed	The operator is not allowed to use the command, or he is not operator for the device referenced (error).
no resources	PRIMO is not able to execute the command because of missing resources (error).
receiver entry troubles	PRIMO is not able to execute the 'route' command because the entry describing the destination does not exist or is inconsistent (error).

receiver device troubles

PRIMO is not able to execute the 'route' command because of troubles with the device stated in the destination (error).

killed by application

PRIMO is not able to execute the 'route' command because the transport has been killed by the application (error).

end of area

End of medium encountered during skip action.

```
! date of options                                    ! options      :=   0,
! number of printer coroutines                      ! prcount      :=   1,
! size of printer buffer (halfwords)                ! prbufsize    := 128,
! leading and trailing page on printer lists        ! prltpage     :=   1,
! max lines pr printer page                         ! prlinepage   := 100,
! number of punch coroutines                        ! pccount      :=   1,
! size of punch buffer (halfwords)                  ! pcbufsize    := 128,
! number of reader coroutines                       ! rdcount      :=   1,
! size of reader buffer (halfwords)                 ! rdbufsize    := 128,
! number of cardreader coroutines                   ! cdcount      :=   0,
! size of cardreader buffer (halfwords)             ! cdbufsize    := 108,
! number of tty coroutines (halfwords)              ! twcount      :=   0,
! size of tty buffer                                ! twbufsize    := 104,
! number of format8000 printer coroutines           ! fprcount     :=   0,
! size of format8000 printer buffer (hlfwords)      ! fprbufsize   := 172,
! no of operator coroutines                         ! oprcount     :=   2,
! no of transport description segmnts               ! trsegm       := 100,
! size of testoutput area                           ! testsegmnts  :=  42,
! transport description save period                 ! trsaveminut  := 240,
! no of waiting transports  ( total )               ! waittrans    :=  50,
! no of pending wait operations                     ! waitops      :=   5,
! output details to operator ( when <> 0)          ! oprdetails   :=   0,
```

PRIMO Resource Consumption

In the go through below the resource demands when running PRIMO are listed.

As it may be seen from the formulas, the demands vary much depending on the trimming of PRIMO.

Primary store (halfwords):

Constant consumption approx.:	10100
Operator coroutines:	oprcount * 180
Printer coroutines:	prcount * (700 + prbufsize)
Punch coroutines:	pccount * (680 + pcbufsize)
Reader coroutines:	rdcount * (680 + rdbufsize)
Card reader coroutines:	cdcount * (680 + cdbufsize)
Type writer coroutines:	twcount * (680 + twbufsize)
IBM 3270 printer coroutine:	fprcount * (780 + fprbufsize)
Transport queue elements:	waittrans * 10
Testbuffer (optional)	1024

Message buffers:

Constant consumption	2
Operator coroutines	oprcount
Printer coroutines	prcount
Punch coroutines	pccount
Reader coroutines	rdcount
Card reader coroutines	cdcount
Type writer coroutines	twcount
IBM 3270 printer coroutines	fprcount
"Wait and get state" operations	waitops

Area processes:

Constant consumption	3
Printer coroutines	prcount
Punch coroutines	pccount
Reader coroutines	rdcount
Card reader coroutines	cdcount
Type writer coroutines	twcount
IBM 3270 printer coroutines	fprcount

BS segments:

Description area	trsegm
Test area	testsegments

(Note that because of slice truncation one extra segment slice must be allocated).

Example:

With the standard trimming defined in appendix C, the resource demands will be:

Primary store (halfwords)	14400
Message buffers	12
Area processes	6
BS segments	142 + 1 slice

RETURN LETTER

Title: PRIMO (4th Edition)
User's Guide/Reference Manual/
Installation Guide

RCSL No.: 31-D688

A/S Regnecentralen af 1979/RC Computer A/S maintains a continual effort to improve the quality and usefulness of its publications. To do this effectively we need user feedback, your critical evaluation of this manual.

Please comment on this manual's completeness, accuracy, organization, usability, and readability:

Do you find errors in this manual? If so, specify by page.

How can this manual be improved?

Other comments?

Name: _____ Title: _____

Company: _____

Address: _____

Date: _____

Thank you

42-i 1288

..... Fold here .....

..... Do not tear - Fold here and staple .....

Affix
postage
here

REGNECENTRALEN
af 1979
Information Department
Lautrupbjerg 1
DK-2750 Ballerup
Denmark