
RCSL No: 31-D557

Edition: March 1979

Author: Henrik Quistgaard
Jens Ramsbøl

Title:

CREATELINK, RELEaselink, LOOKUPLINK, LOOKUPDEV
UTILITY PROGRAMS

Keywords:

RC4000, RC8000, RCNET, link, jobhost, devicehost, utility programming documentation.

Abstract:

This manual describes 4 utility programs used for creating, releasing, and looking up links from an RC4000/RC8000 jobhost to peripheral devices on a devicehost in RCNET.

(50 printed pages)

Copyright © 1981, A/S Regnecentralen af 1979
RC Computer A/S

Printed by A/S Regnecentralen af 1979, Copenhagen

Users of this manual are cautioned that the specifications contained herein are subject to change by RC at any time without prior notice. RC is not responsible for typographical or arithmetic errors which may appear in this manual and shall not be responsible for any damages caused by reliance on any of the materials presented.

CONTENTS	PAGE
1. INTRODUCTION	1
1.1 Jobhost and Devicehost	1
1.2 Links	2
1.3 Device Qualifications	3
1.4 Communication Flow at a Jobhost	4
2. CREATELINK	7
2.1 Example	7
2.2 Syntax	8
2.3 Semantics	9
2.4 Function	11
2.4.1 Host Statements	11
2.4.2 Parameter Statements	12
2.4.3 Device Statements	13
2.5 Program Output	15
2.6 Standard Values of the Device Parameters	15
2.7 Survey on the Device Names at a Devicehost	17
3. RELEASELINK	19
3.1 Example	19
3.2 Syntax	19
3.3 Semantics	19
3.4 Function	20
3.5 Program Output	20
4. LOOKUPLINK	21
4.1 Syntax	21
4.2 Semantics	21
4.3 Function	21
4.4 Program Output	22
5. LOOKUPDEV	24
5.1 Syntax	24
5.2 Semantics	24
5.3 Function	24
5.4 Program Output	25

CONTENTS (continued)	PAGE
----------------------	------

6. EXAMPLES	27
7. USE OF THE PROGRAMS UNDER DIFFERENT OPERATING SYSTEMS ...	31
7.1 Operating System s	31
7.2 Operating System BOSS	32
7.3 Operating System SOS	33
8. ERROR MESSAGES	34

APPENDICES

A. REFERENCES	40
B. PROTECTION SYSTEM FOR PROCESSES AND CATALOG ENTRIES	41

1. INTRODUCTION.

1.

This manual describes 4 utility programs used for creating, releasing, and looking up links to peripheral devices on a devicehost in RCNET.

The programs fully replace their predecessor, the LINKCENTRAL program.

The 4 programs are described in chapter 2-5, examples are found in chapter 6, and error messages from the programs are gathered in chapter 8. Chapter 7 deals with special circumstances, when running the programs under different operating systems. Appendix B holds a summary of the protection system in connection with catalog entries and processes as the user of the link programs must be familiar with this system.

The rest of this chapter explains some of the concepts used in connection with links.

1.1 Jobhost and Devicehost.

1.1

Communication according to the RCNET - Device Control Protocol [1] takes place between a host executing a userjob, denoted the Jobhost, and a device at another host, denoted the Devicehost.

A host is addressed by its host identification (HOST-ID) and the identification of the network (NET-ID) to which the host belongs.

With respect to the communication the jobhost will be the master initiating operations at the device, which returns answers when the operations have been carried out.

At the present state of RCNET, RC4000/RC8000 can act as jobhost and frontend computers or concentrators like RC3600/RC3500/DP2200 and compatible models as devicehosts.

The term device refers either to a physical device or to an application program "playing" the role of a device through the interface defined in [2].

1.2 Links.

1.2

The communication is carried out on a so-called link. This is a logical connection through RCNET between the jobhost and the device situated at a devicehost or more precisely, a logical connection between an external process at the jobhost and a device coroutine at the devicehost. In this context the external process is called a link process (see section 1.4). At the jobhost the link serves as a short way of referencing the device after the creation of the link.

The link also serves as a reservation mechanism, as a device may only be occupied by one link at a time.

A link at a host is identified by its linkno, used as a logical reference. At a jobhost the linkno will always be equal to the logical device number in the monitor.

The total identification of a link consists of

- identification of devicehost
- linkno at devicehost
- identification of jobhost
- linkno at jobhost

A link can be of either permanent or temporary type. The differences between these types are:

permanent link

The link is created, at the jobhost, to a particular logical device number that means to a particular link process. After set up of the link, the creating process and all of its ancestors are included as users of the device. In case of severe faults in the

connection between job- and devicehost, the link process is cleaned (all messages in queue are returned with result 4, malfunction), but the link is not removed.

temporary link

The link at the jobhost is created to a free link process, which means that the logical device number cannot be decided by the user. Only the creating internal process is included as a user of the device. A consequence of this is that the link is released when the process is removed. In case of severe faults in the connection between job- and devicehost the link process is cleaned and the link is removed.

A link is dynamically created and removed. The initiative for creation may derive from the jobhost, the devicehost (activation of the attention-key on a terminal), or from a third host.

Normally temporary links are used in connection with terminals and permanent links in connection with other devices.

If RC8000 functions as jobhost the permanent links can be defined in the options to the monitor, meaning that the links automatically will be created in the start-up phase. When RC4000 functions as jobhost the links must be created after the system is initialized by help of the CREATELINK program. In both cases links can later on be created and released.

1.3 Device Qualifications.

1.3

In order to initiate creation of a link to a certain device the jobhost must set up a number of qualifications that the device should possess.

This information is then passed on to the devicehost, which, if possible, selects the device. When the devicehost returns an answer, it will include the actual qualifications possessed by the device se-

lected.

The operation includes the following device characteristics:

Devicename.

The internal name of the device at the devicehost (do not confuse this name with the name of the link process at the jobhost).

The devicename is fixed and cannot be changed by the user.

Kind.

Is the type-identification of the device intended to be used for coding of functional characteristics.

Mode.

Specifies that the device should be able to operate in a certain mode, e.g. high density on a magnetic tape.

The mode parameter is not relevant in link creation.

Buffersize.

The desired size of datablocks that can be transferred at a time.

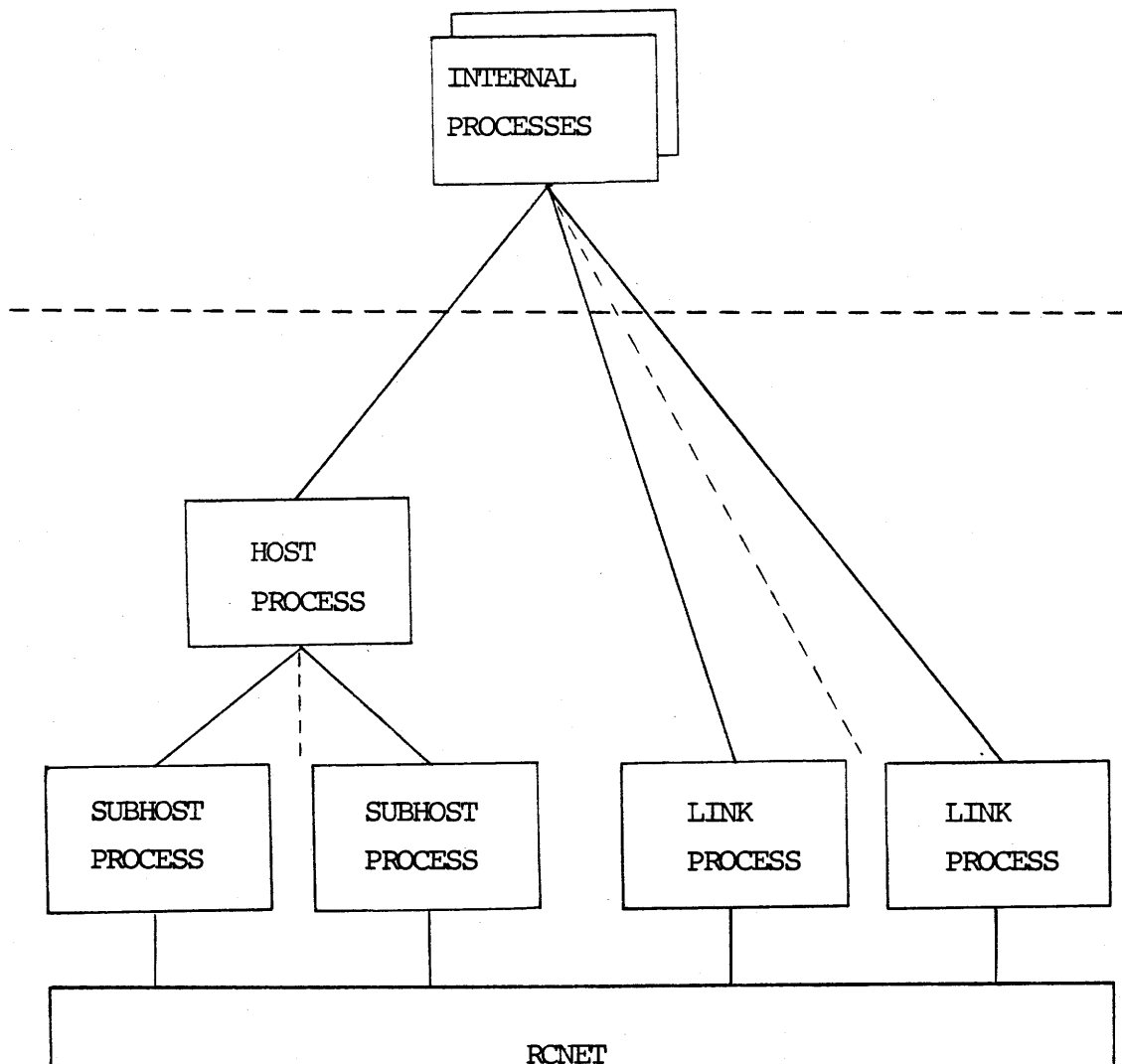
Buffers.

The desired number of buffers, which at the same time can be handled at the devicehost.

1.4 Communication Flow at a Jobhost.

1.4

To facilitate the understanding of the function of the utility programs and of some of the error messages the communication flow at a jobhost is illustrated in the following figure.



Figur no 1. Communication Flow at a Jobhost.

Communication is carried out by means of operations. An operation consists of a message initiating, and an answer telling the result of an action. Two types of operations exist: supervisor and normal operations. Supervisor operations control the creation and removal of links, and are sent to the host process.

The main task of the host process is to determine the subhost process to handle the received supervisor operation.

One subhost exists for each connected Frontend Processor Adaptor (FPA, RC8000) or Synchronous Communication Controller (SCC, RC4000).

The host process only accepts messages from internal processes,

which are users of the specified subhost process.

Normal operations are used after link creation, to initiate actions at the device, such as input or output of data, and are sent to the link process linked up to the device.

The program running in a link process is denoted a link driver, which acts as the interface between formats used by jobs when communicating with the devices and the formats used within the Device Control Protocol.

All programs below the dot-and-dash line are included in the monitor.

2. CREATELINK.

2.

The utility program CREATELINK is used for setting up links to peripheral devices connected to devicehosts in RCNET. A link can be of either permanent or temporary type.

2.1 Example.

2.1

The command

```
createlink host.ba08 lp.printer2.temp.name.pr03
```

will establish a temporary link to the device with name 'printer2' at the devicehost defined in the catalog entry named 'ba08'. After the link creation the link process is named 'pr03'.

2.2 Syntax.

2.2

$$\{ \langle \text{outfile} \rangle = \}_0^1$$

$$\text{createlink } \langle \text{hostdef} \rangle \left[\left[\langle \text{param def} \rangle \right]_0^* \left[\langle \text{device def} \rangle \right]_1^* \right]_1^*$$

$$\langle \text{hostdef} \rangle ::= \left\{ \begin{array}{l} \text{host. } \langle \text{subhostno} \rangle . \langle \text{host-id} \rangle \left[. \langle \text{net-id} \rangle \right]_0^1 \\ \text{host. } \langle \text{hostname} \rangle \\ \text{hostref. } \left\{ \begin{array}{l} \langle \text{logical devno} \rangle \\ \langle \text{process name} \rangle \end{array} \right\} \end{array} \right\}$$

$$\langle \text{param def} \rangle ::= \left\{ \begin{array}{l} \text{bufsize. } \left\{ \begin{array}{l} \langle \text{buffersize} \rangle \\ \text{max} \\ \text{std} \end{array} \right\} \\ \text{buffers. } \left\{ \begin{array}{l} \langle \text{no of buffers} \rangle \\ \text{max} \\ \text{std} \end{array} \right\} \end{array} \right\}$$

$$\langle \text{device def} \rangle ::= \langle \text{modekind} \rangle . \langle \text{devname} \rangle . \left\{ \begin{array}{l} \langle \text{devno} \rangle \\ \text{temp} \end{array} \right\} \left\{ \begin{array}{l} .\text{name}. \langle \text{procname} \rangle \\ .\text{entry}. \langle \text{entry name} \rangle \end{array} \right\}_0^1$$

$$\langle \text{modekind} \rangle ::= \left\{ \begin{array}{l} \text{gac} \\ \text{bs} \\ \text{tw} \\ \text{tro} \\ \text{tre} \\ \text{trn} \\ \text{trf} \\ \text{tpo} \\ \text{tpe} \\ \text{tpn} \\ \text{tpf} \\ \text{lp} \\ \text{crb} \\ \text{crc} \\ \text{cmb} \\ \text{cmd} \\ \text{crba} \\ \text{mto} \\ \text{mte} \\ \text{nrz} \\ \text{nrze} \\ \text{pl} \\ \text{flx} \\ \text{tlx} \end{array} \right\}$$

<outfile> The name of a bs-file or a catalog entry describing a peripheral device. If an **<outfile>** is present in call this file is used for program output including possible error messages - otherwise the current output file is used. If **<outfile>** is not found in the catalog a bs-file with scope temp is created on the main-disc.

<subhostno> Subhostnumber, the logical device number at the jobhost of the subhost process, which takes care of the communication to the devicehost. Each FPA (SCC) has an adjacent subhost process.

<host-id> Devicehost identification, a unique identification, denoted HOST-ID, of the host within the network. At the present state of RCNET in context with the Device Control Protocol, the HOST-ID is an unambiguous identification of a devicehost.

<netid> Network identification. In order to be prepared for connection of networks, the HOST-ID is extended with an identification, denoted NET-ID, of the network to which the devicehost belongs. At present the parameter is not used and can be omitted in the program call (default value = 0).

<hostname> Name of a catalog entry that describes the devicehost.

The catalog entry must be created like:

<hostname> = set ip host d.0 **<subhostno>** **<host-id>**
<net-id>

e.g.

ba08 = set ip host d.0 17 5012 0

<logical devno>	Logical device number of a link process linked up to some device at the wanted devicehost.
<process name>	Process name of a link process linked up to some device at the wanted devicehost.
<buffersize>	Desired size in 8 bit characters of datablocks. The parameters "max" and "std" are explained in the next section.
<no of buffers>	Desired number of buffers.
<modekind>	Specifies the mode and kind of the device. Mode is only used when a catalog entry has to be created describing the device (see parameter <entry name>).
<devname>	Internal name of the device at the devicehost.
<devno>	Logical device number in the monitor of the link process to which a <u>permanent</u> link shall be created.
	When the text "temp" is used the link is created as a temporary link - in which case the device number is selected by the monitor as the number of a free link process.
<proc name>	Specifies that the link process after creation of the link should be named with this name.
	If the parameter is omitted the link process is given a work name "subxxx" by the monitor, where xxx is a number in the interval 000 <= xxx <= 999.

<entry name> Specifies that a catalog entry with this name describing the device should be created after creation of the link. The entry tail will hold the modekind and as document name the name of the link process generated by the monitor.

2.4 Function.

2.4

The parametergroups, hereafter called statements, in the program can be divided into three types - host, parameter, and device statements.

The host and parameter statements define the address of the device-host in the network and some of the device parameters. A device statement creates a link to a device, either a permanent or a temporary one.

The different statements are described in the following sections.

2.4.1 Host Statements.

2.4.1

This statement defines the devicehost, to which the links shall be connected, and must always be the first statement in the program call.

The statement can have the formats:

```
host. <subhostno> . <host-id>
host. <subhostno> . <host-id> . <net-id>
host. <hostname> .
hostref. { <logical devno>
           <process name> }
```

The two first formats are used when direct addressing is wanted. An indirect addressing is used in the last two formats, specifying either a catalog entry describing the devicehost or a link process

linked up to some device at the devicehost.

More details on the parameters are found in section 2.3.

2.4.2 Parameter Statements.

2.4.2

To each device kind is associated standard values for the device parameters `<buffersize>` and `<buffers>` used when a link is created.

These standard values can, however, be changed by parameter statements of the formats:

$$\begin{array}{l} \text{bufsize.} \quad \left\{ \begin{array}{l} \text{<buffersize>} \\ \text{max} \\ \text{std} \end{array} \right\} \\ \\ \text{buffers.} \quad \left\{ \begin{array}{l} \text{<no of buffers>} \\ \text{max} \\ \text{std} \end{array} \right\} \end{array}$$

If 'max' is specified the parameter is assigned the maximum value that means to the biggest value supported by the basic software at the devicehost. These maximum values can be examined by means of the LOOKUPDEV program (see chapter 5).

After a parameter statement all the following device statements will - independent of kind - use the new value in the link creation.

If you want to return to the standard value associated to the kind, this is performed by specifying 'std' as the second parameter.

The standard values of the device parameters are found in section 2.6.

2.4.3 Device Statement.

2.4.3

The function of a device statement is to establish a link to a device at the devicehost defined in the preceeding host statement.

The statement can have the following formats:

$$\begin{aligned} &\langle \text{modekind} \rangle . \langle \text{devname} \rangle . \left\{ \begin{array}{l} \langle \text{devno} \rangle \\ \text{temp} \end{array} \right\} \\ &\langle \text{modekind} \rangle . \langle \text{devname} \rangle . \left\{ \begin{array}{l} \langle \text{devno} \rangle \\ \text{temp} \end{array} \right\} . \left\{ \begin{array}{l} \text{name.} \langle \text{procname} \rangle \\ \text{entry.} \langle \text{entryname} \rangle \end{array} \right\} \end{aligned}$$

The parameter `modekind` is in connection with the link creation used only to determine the standard values of the device parameters `<buffersize>` and `<buffers>`. The meaning of the different `<modekind>` parameters and the corresponding integer values of mode and kind are described in section 2.6.

`<devname>` points out a device, or more precisely a physical channel, e.g. a multiplexer channel, at the devicehost. The devices are named dependent of kind and the set of device names is defined in the options for the basic system software corresponding to the actual configuration.

An exception is valid if `<modekind>` specifies `telex`. The interface to telex equipment is software-wise implemented so that the `<devname>` is a dummy parameter, which should only be included of syntactical reasons.

Section 2.7 holds a survey of the possible device names.

The next parameter can either be a number or the text "temp". In the first case a permanent link is created to the link process associated with the logical device number `<devno>`. In the second case a temporary link is created.

The link process is associated with the max interval base.

A specific name can be given to the link process after the link creation, if name.<procname> is added to the call. The CREATELINK program performs the renaming by means of the privileged monitor procedure "create peripheral process", and this is only allowed if bit no. 4 is included in the function mask of the calling process (see ref. [3] and [4]). It is important to know that "create peripheral process" cuts down the base of the link process to the catbase (viewpoint base) of the calling process.

It is possible to get hold of the workname generated by the monitor, when no name is given explicitly, by terminating the call with entry. <entryname>. In this case a temporary catalog entry, with entry base equal to catbase of the calling process, is created, containing <modekind> and the workname.

A link will always be created with the smaller of the values of the device parameter <bufsize> specified in the program call (standard or explicitly given) and the maximal value supported by the basic software at the devicehost. The same is valid for the device parameter <buffers>.

The create permanent link operation may possibly cause one or two other links to be released as illustrated in the following table:

state of device at device- host state of link process at jobhost	no link	link present
free	1)	3)
link present	2)	4)

- 1) No links are released.
- 2) The existing link to the link process is removed, meaning that the connection to another device is disconnected.
- 3) The existing link to the addressed device, possibly connected to a link process at another jobhost, is released.
- 4) The existing links to the link process and to the addressed device (may be the same) are released.

The create temporary link operation will not cause any links to be released. If the addressed device occupies a temporary link the operation is dummy and if a permanent link is present the operation is rejected with an error message.

The possibilities and limitations of the link utility programs under different operating systems are described in chapter 7.

2.5 Program Output.

2.5

In addition to possible error messages the program outputs the following terminating line:

end links created: <number of links created>

2.6 Standard Values of the Device Parameters.

2.6

Listed below are the interpretation of the <modekind> parameter and corresponding standard values of the device parameters <bufsize> and <buffers>.

<u><modekind></u>	<u>mode</u>	<u>kind</u>	<u>bufsize</u>	<u>buffers</u>	<u>use of the mode-kind</u>
gac	0	0	10000 ¹⁾	1	General application corou- tine interface to an ap- plication program.
bs	0	6	768	1	Disc
tw	0	8	86	1	Terminal
tro	0	10	258	1	Tape reader, odd parity.
tre	2	10	258	1	Tape reader, even parity.
trn	4	10	258	1	Tape reader, no parity.
trf	6	10	258	1	Tape reader, flexo code.
tpo	0	12	258	1	Tape punch, odd parity.
tpe	2	12	258	1	Tape punch, even parity.
tpn	4	12	258	1	Tape punch, no parity.
tpf	6	12	258	1	Tape punch, flexo code.
lp	0	14	258	1	Line printer
crb	0	16	258	1	Card reader, binary.
crc	10	16	258	1	Card reader, EBCDIC.
cmb	64	16	258	1	Card reader, marksense binary.
cmd	74	16	258	1	Card reader, marksense decimal.
crba	256	16	258	1	Card reader, BASIC format.
mta	0	18	3090 ²⁾	2	Mag tape, odd parity, normal.
mtb	2	18	3090	2	Mag tape, even parity, normal.
nrz	4	18	3090	2	Mag tape, odd parity, special.
nrze	6	18	3090	2	Mag tape, even parity, special.
pl	0	20	258	1	Plotter
flx	0	22	513	2	Flexible disc
tlx	0	36	258	1	Telex device

1) The buffersize is assigned to the size specified in application program description (APDES).

2) Number of characters on 4 segments on a disc + 18 required by the SAVE/LOAD utility programs.

2.7 Survey on the Device Names at a Devicehost.

2.7

The devices are named dependent on kind, possibly followed by a number.

Terminals: TERMINAL1 1)
 TERMINAL2
 ...
 ...

Possible telex devices:

Line printers: PRINTER1
 PRINTER2
 ...
 ...

Card readers: CARDREADER
 CARDREADER1
 ...
 ...

Tape readers: READER
 READER1
 ...
 ...

Tape punches: PUNCH
 PUNCH1
 ...
 ...

Plotters: PLOTTER
 PLOTTER1
 ...
 ...

Magtape stations: MT0
 MT1
 ...
 ...

Cassette tape units: CT0
 CT1
 ...
 ...

Disc drives: DISC0
 DISC1
 ...
 ...

Flexible discs:

FD0
FD1
...
...

Possible application programs:

HDLC channels:

INHDL1
INHDL2
...
...

OUTHDL1
OUTHDL2
...
...

- 1) If a main console (TTY) is present it always has the name
 TERMINAL.

3. RELEASELINK.

3.

The program RELEASELINK is used to remove links to devices in RCNET.

The links can be temporary as well as permanent.

3.1 Example.

3.1

The command

```
releaselink 27 31 printer1 sub008
```

will cause the link processes named "printer1" and "sub008", as well as the processes with the device numbers 27 and 31 to be removed.

3.2 Syntax.

3.2

$$\{\text{<outfile>}\}_{0}^{1} \text{ releaselink } \left\{ \begin{array}{l} \text{<devno>} \\ \text{<process name>} \\ \text{<entryname>} \end{array} \right\}_{1}^{*}$$

3.3 Semantics.

3.3

<outfile> See section 2.3

<devno> Logical device number of a link process.

<entry name> Name of a non-area catalog entry holding the name of a link process as the document name.

3.4 Function.

3.4

Each parameter defines a link process connected to a link that shall be released.

A link process is specified either by its logical device number or by the process name, which can be implicitly given through the name of a catalog entry.

In case a process is defined by an entry name the entry is removed too. If the calling process is not allowed to remove the entry, the link will remain untouched.

3.5 Program Output.

3.5

In addition to possible error messages the program outputs the following terminating line:

end links released: <number of links released>

4. LOOKUPLINK.

4.

The program LOOKUPLINK prints all parameters, except the device name, describing a link in RCNET.

4.1 Syntax.

4.1

$$\{ \langle \text{outfile} \rangle = \} \begin{matrix} 1 \\ 0 \end{matrix} \quad \text{lookuplink} \quad \left\{ \begin{matrix} \left\{ \begin{matrix} \langle \text{devno} \rangle \\ \langle \text{process name} \rangle \\ \langle \text{entry name} \rangle \end{matrix} \right\}^* \\ \text{all} \end{matrix} \right\}_1$$

4.2 Semantics.

4.2

$\langle \text{outfile} \rangle$	See section 2.3
$\langle \text{devno} \rangle$	Logical device number of a link process.
$\langle \text{process name} \rangle$	Name of a link process.
$\langle \text{entry name} \rangle$	Name of a non-area catalog entry holding the name of a link process as the document name.

4.3 Function.

4.3

Each parameter defines a link process, whose parameters relevant for link description, are to be printed.

If "all" is specified a complete survey of all the link processes at the jobhost is printed.

4.4 Program Output.

4.4

A link survey consists of a heading and 2 lines for each link process selected by the command. The heading holds a survey identity and date/time of the survey generation. The next line shows an example of a link survey command:

```
lookuplink 12 printer1 24
```

The resulting printout is shown in the following 10 lines.

```
LINK SURVEY D.790223.1118
```

IDENT	KIND	BUF	SIZE	PROC.NAME	JH.LNO	JH.ID	JH.NID
TYPE				SUBHOSTNO	DH.LNO	DH.ID	DH.NID
12	8	1	172	SUB028	12	105	0
TEMP				17	23	5013	0
PRINTER1	14	2	258	PRINTER1	16	105	0
PERM				18	4	5014	0

```
24
```

```
FREE
```

```
END LINKS LOOKED UP: 3
```

```
aaaa  bb  c  dddd  eeee  fff  gggg  hhh
iiii          jj  kkk  1111  mmm
```

The 2 lines above show the survey layout and the contents is explained through the terms of the layout.

<u>Term</u>	<u>Contents</u>
aaaa	Logical device number or name of the link process as specified in the command.
bb	Device kind.
c	Number of buffers.
dddd	Buffersize.
eeee	Name of the link process.
fff	Jobhost linkno, always equal to the logical device number.
gggg	Jobhost identification.
hhh	Jobhost net-id.
iiii	Type of link. TEMP: Temporary link PERM: Permanent link FREE: Free link process
jj	Subhostno, the logical device number of the subhost process.
kkk	Devicehost linkno. To each device (device name) is associated a fixed linkno at the devicehost.
llll	Devicehost identification.
mmm	Devicehost net-id.

5. LOOKUPDEV.

5.

The program LOOKUPDEV prints all device qualifications for a device at a devicehost as given in the options, when the basic system software was generated for the actual installation.

5.1 Syntax.

5.1

$$\left\{ \langle \text{outfile} \rangle = \right\} \begin{matrix} 1 \\ 0 \end{matrix} \quad \text{lookupdev} \quad \langle \text{hostdef} \rangle \left\{ \begin{matrix} \left\{ \text{dev} . \langle \text{devname} \rangle \right\}^* \\ \text{all} \end{matrix} \right. \begin{matrix} 1 \\ 1 \end{matrix} \right\}$$

$$\langle \text{hostdef} \rangle :: = \left\{ \begin{matrix} \text{host} . \langle \text{subhostno} \rangle . \langle \text{host-id} \rangle & \left\{ . \langle \text{netid} \rangle \right\}^* \\ \text{host} . \langle \text{hostname} \rangle & \\ \text{hostref} . \left\{ \begin{matrix} \langle \text{logical devno} \rangle \\ \langle \text{process name} \rangle \end{matrix} \right\} & \end{matrix} \right. \begin{matrix} 1 \\ 0 \end{matrix} \right\}$$

5.2 Semantics.

5.2

$\langle \text{outfile} \rangle$	See section 2.3
$\langle \text{hostdef} \rangle$	Defines the devicehost as explained in section 2.3.
$\langle \text{devname} \rangle$	Internal name of a device at the devicehost.

5.3 Function.

5.3

For each defined device a survey is printed holding the device parameters corresponding to the options for the basic software at the specified devicehost.

If "all" is specified a survey of all devices at the devicehost is printed. However, the total survey will not include telex devices or devices connected via the GAC-interface, except of flexible disc units.

5.4 Program Output.

5.4

A device survey consists of a heading and 1 line for each device selected by the command. The heading holds a survey identity and date/time of the survey generation. The next line shows an example of a device survey command.

```
lookupdev host.17.5013 dev.terminal7 dev.printer1
```

The resulting printout is shown in the following 5 lines.

```
DEVICE SURVEY D.790223.1721
```

```
DEVICE NAME    KIND  BUF  SIZE  DH.LNO  HOSTNO  DH.ID  DH.NID
```

```
TERMINAL7      8    1   86    6      17    5013    0
```

```
PRINTER1      14    1  258   18      17    5013    0
```

```
END DEVICE LOOKED UP: 2
```

```
aaaa          bb    c    dddd eee    ff      gggg   hhh
```

The line above shows the survey layout and the contents is explained through the term in the layout.

<u>Term</u>	<u>Contents</u>
aaaa	Device name at the devicehost.
bb	Device kind. The value of kind will be printed as 255 if no link is connected to the device.
c	Number of buffers.
dddd	Buffersize.

eee Devicehost linkno.

The value of linkno corresponds with the succession of devices found in section 2.7, and with linkno 0 being the first value associated with "terminal1". Due to limitation in the implemented supervisor operations mentioned in section 1.4 the linkno can only be stated for

- 1) terminals
- 2) other devices if "all" is specified, except devices/application programs listed after flexible discs in the survey in section 2.7.

ff Subhostno, the logical device number of the subhost process.

gggg Devicehost identification.

hhh Devicehost net-id.

6. EXAMPLES.

6.

The following theoretical examples illustrate the function of the different programs.

At a devicehost (subhostnumber, host-id = 21, 6543) permanent links should be established to a mag tape station and to a cardreader and a temporary link to a line printer.

The links must possess the following qualifications:

	<bufsize>	<buffers>	<devno>	<name>
Mag tape	3090	2	3	-
Cardreader	120	1	5	card1
Line printer	258	2	-	1)

- 1) The generated name of the link process should be saved in a catalog entry named lp2.

Furthermore it is known that the link process "console1" is linked up to a terminal on the devicehost in question.

The situation at the jobhost and the devicehost is illustrated by means of the LOOKUP commands.

```
lookuplink 0 1 2 3 4 5
```

LINK SURVEY D.790227.1348

IDENT TYPE	KIND	BUF	SIZE	PROC.NAME SUBHOSTNO	JH.LNO DH.LNO	JH.ID DH.ID	JH.NID DH.NID
0 TEMP	8	1	86	SUB012 21	0 2	105 6543	0 0
1 FREE							
2 PERM	8	1	86	CONSOLE1 21	2 0	105 6543	0 0
3 FREE							
4 PERM	22	2	513	FD0 21	5 9	105 6543	0 0
5 PERM	12	1	258	PUNCH 21	6 7	105 6543	0 0

END LINKS LOOKED UP: 6

lookupdev hostref.console1 all

DEVICE SURVEY D.790227.1349

DEVICE NAME	KIND	BUF	SIZE	DH.LNO	DH.ID	DH.NID
TERMINAL1	8	1	86	0	6543	0
TERMINAL2	255	1	86	1	6543	0
TERMINAL3	8	1	86	2	6543	0
TERMINAL4	255	1	86	3	6543	0
PRINTER	255	2	258	4	6543	0
CARDREADER	255	1	258	5	6543	0
READER	255	1	258	6	6543	0
PUNCH	12	1	258	7	6543	0
MT0	18	2	3090	8	6543	0
FD0	22	2	513	9	6543	0

END LINKS LOOKED UP: 10

It is assumed that the mag tape station is linked up to another job-host in the network.

Now the links are created by:

```
createlink host.21.6543 mt.mt0.3,
          bufsize.120 crb.cardreader.5.name.card1,
          bufsize.std buffer.2 lp.printer.temp.entry.lp2
```

The same two lookup commands as before will now generate the following output:

LINK SURVEY D.790227.1354

IDENT TYPE	KIND	BUF	SIZE	PROC.NAME SUBHOSTNO	JH.LNO DH.LNO	JH.ID DH.ID	JH.NID DH.NID
0 TEMP	8	1	86	SUB012 21	0 2	105 6543	0 0
1 TEMP	14	2	258	SUB015 21	1 4	105 6543	0 0
2 PERM	8	1	86	CONSOLE1 21	2 0	105 6543	0 0
3 PERM	18	2	3090	SUB014 21	3 8	105 6543	0 0
4 PERM	22	2	513	FD0 21	4 9	105 6543	0 0
5 PERM	16	1	120	CARD1 21	5 5	105 6543	0 0

END LINKS LOOKED UP: 6

DEVICE SURVEY D.790227.1354

DEVICE NAME	KIND	BUF	SIZE	DH.LNO	DH.ID	DH.NID
TERMINAL1	8	1	86	0	6543	0
TERMINAL2	255	1	86	1	6543	0
TERMINAL3	8	1	86	2	6543	0
TERMINAL4	255	1	86	3	6543	0
PRINTER	14	2	258	4	6543	0
CARDREADER	16	1	258	5	6543	0
READER	255	1	258	6	6543	0
PUNCH	255	1	258	7	6543	0
MT0	18	1	3090	8	6543	0
FD0	22	2	513	9	6543	0

END LINKS LOOKED UP: 10

The entry lp2 will have the contents:

```
lp2 = set lp sub015 0 0 0 0 0; temp  
      ; entry head
```

If it is wanted to release the link established to the flexible disc,
the command

```
releaselink 4
```

is entered and the corresponding survey lines will now look like:

```
4  
FREE
```

and

```
FD0      255      2      513      9      6543      0
```

7. USE OF THE PROGRAMS UNDER DIFFERENT OPERATING SYSTEMS.

7.

This chapter describes how the following limitations and properties in the link programs are handled under the operating systems and BOSS:

- 1) The calling process must be user of the involved subhost process (valid for all 4 programs).
- 2) Renaming of a link process requires that bit 4 (create peripheral process) is included in the function mask of the calling process (CREATELINK).
- 3) The base of a link process is changed from the max. interval to catbase of the calling process, when the link process is renamed (CREATELINK).

As explained below the link programs cannot be used under the operating system SOS.

7.1 Operating Systems.

7.1

- Ad 1) When a process is created under the operating systems, the process will be indicated as a user of all "devices", including the existing subhost processes.

In this context it should be noticed that when a permanent link is created the calling process and all of its ancestors are included as users of the device. In other words only processes created after the link is established and not existing processes are included as users.

See also the description of the s-commands INCLUDE and EXCLUDE in ref. [4].

- Ad 2) The privilege is obtained by means of the s-command FUNCTION or by assigning the function field in the actual entry in the s-user-catalog.

Ad 3) When an internal process is created the catbase is set to the standard base.

The standard base and with this the catbase of a process can be specified either in the actual entry in the s-user-catalog or by means of the s-commands BASE and LOGIN.

Furthermore it is emphasized that the s-command CALL will extend the base of the corresponding link process to the max. interval.

7.2 Operating System BOSS.

7.2

Ad 1) A BOSS user is included as a user of a subhost process if device <logical device number of subhost process> is specified in the job specification.

Subhost "devices" must be specified in BOSS options and it is recommended that they are specified as devices without a "mnemonic name" (the list after e23). This means that the operator will be warned before the user is going to communicate with the subhost "device". If the warning is not wanted the subhost "devices" should be specified after the list e25 (see ref. [6]).

It is possible, but troublesome, to exclude projects and users from using a subhost "device" by the following specification in the user catalog:

6 device no

6 device <list of allowed devices>

Finally it should be mentioned that temporary links also can be created by means of the parent message create link.

Ad 2) The privilege cannot be obtained under BOSS. However, the renaming can be performed by means of the operator command:

call <devno> <name>

Furthermore the attention is drawn to the entry. <entryname>

feature in the call of CREATELINK, where the user can get access to the internal name generated by the monitor.

Ad 3) As renaming is not allowed the link process will always be associated to the max. interval base.

7.3 Operating System SOS.

7.3

SOS does not allow a user to be user of a subhost process, which means that is is impossible to run the link programs under this operating system.

8. ERROR MESSAGES.

8.

Error messages from the programs all have the format:

***<program name> <number> <error cause>

<number> is the number of the statement (parameter/parametergroup), where the error is detected, counted so that the first statement after the program name has no 1.

The possible error causes are listed below in alphabetic order together with an indication of the programs in which the error can occur.

CL: CREATELINK
 RL: RELEASELINK
 LL: LOOKUPLINK
 LD: LOOKUPDEV

Normally a program continues after an error message, if not it is explicitly stated in the explanation to the message. In all cases the execution is terminated by setting the fp modebit ok.no.

CL	RL	LL	LD	
----	----	----	----	--

X	X	X	X	call
---	---	---	---	------

No right-hand parameters present.

X	X	X	X	catalog error
---	---	---	---	---------------

Catalog input/output error. The program is terminated in case of CREATELINK and LOOKUPDEV.

X				claims exceeded
---	--	--	--	-----------------

No resources for creating the specified catalog entry.

CL	RL	LL	LD
----	----	----	----

X	X	X	X	connect outfile impossible
---	---	---	---	----------------------------

Outfile specifies a non-area entry and the corresponding device cannot be connected due to malfunction or reservation. Current output is used.

X				create entry not allowed
---	--	--	--	--------------------------

Catbase of calling process does not allow creation of the specified entry, i.e. catbase of calling process is outside stdbase of calling process.

X				device closed
---	--	--	--	---------------

The link cannot be created, because the device is closed by the operator at the devicehost. The facility is at present not implemented in the basic software.

X			X	devicehost unknown
---	--	--	---	--------------------

The devicehost specification (host-id, net-id) does not address a devicehost connected to the specified subhost. This is caused by a parameter error in the preceeding host statement.

The program is terminated.

X			X	device name unknown
---	--	--	---	---------------------

No device of the specified name is present at the addressed devicehost.

X				device number unknown
---	--	--	--	-----------------------

No external process with the specified logical device number exists.

X				device reserved by another host
---	--	--	--	---------------------------------

The link cannot be established because the device is reserved from another jobhost in the net.

CL RL LL LD

- X device reserved by another user
 The renaming of the link process was re-
 jected because the process has been reser-
 ved by another user.
- X device reserved by another process
 The device is reserved by another process
 at the devicehost.
- X high priority request
 The link cannot be set up, because a devi-
 ce request with higher priority has been
 received.
- X X hostref error
 The hostref parameter in the preceeding
 host statement does not specify a link pro-
 cess connected to a device at a devicehost.
 The program is terminated.
- X X X X illegal format
 A parameter or the format of a statement is
 incorrect.
- X X X line error
 The operation is terminated due to a line
 break somewhere in the connection to the
 devicehost or to a break down of the devi-
 cehost. This error message may also appear
 for the reason explained under the message
 "devicehost unknown".
 The program is terminated.
- X X link unknown
 The specified external process is not a link
 process.

CL RL LL LD

- X name conflict
 The renaming of the link process or the
 creation of a specified entry is not success-
 full due to a name conflict.
- X X no device statement
 No device statement is found in the call.
- X no driver
 The driver corresponding to the specified
 device does not exist at the devicehost.
- X X no host statement
 The first statement after the program name
 must be a host statement.
 The program is terminated.
- X no resources at devicehost
 The link cannot be created due to missing
 resources at the devicehost.
- X no resources at jobhost
 The link cannot be created because - when
 creating a permanent link - the external
 process pointed out by the logical device
 number is not a link process or - when
 creating a temporary link - no free link
 process is available.
- X permanent link exists
 It is not allowed to create a temporary
 link if a permanent link is associated to
 the device in question (see section 2.4).
- X X X X process not user of subhost
 The operation is rejected because the cal-
 ling process is not included as a user of

CL RL LL LD

the subhost process in question.

In case of CREATELINK and LOOKUPDEV the program is terminated.

X X process unknown

The specified process is not an external process.

X renaming not allowed

The renaming of the link process is rejected because the calling process is not privileged to perform a create peripheral process function (see chapter 7).

The created link is released again and perhaps one or two other links have been removed as explained in section 2.4.

X remove entry not allowed

The specified entry is protected, i.e. base of the entry name is not contained in max. base of the calling process.

X reservation rejected

The link is not established because reservation is rejected. This can only occur in connection with application programs at the devicehost (GAC processes).

X X X X sender stopped

The calling process is stopped during the operation.

X X subhost number error

The subhost number is not the logical device number of a subhost process. This is caused by a wrong parameter in the preceeding host statement.

The program is terminated.

CL RL LL LD

X timeout

In connection with the creation of a permanent link an existing link at the specified device number has been released. The release operation is initiated but not accomplished within 10 seconds. The new link is not created.

This error message may for instance occur in connection with link creation to a mag-tape station, where the release operation includes a rewind of the mounted tape.

Action: try again.

X X unknown hostname

The hostname in the host statement is not the name of a catalog entry.

X X unknown statement

The first parameter in a statement is either a number or an unknown text.

A. REFERENCES.

- [1] RCSL No.: 43-Ri0174, June 1974, Erik Lilholt
RCNET, Device Control Protocol (Reference Manual)
- [2] RCSL No.: 43-Ri0260, April 1977, Ole Krag Hansen
RCNET, User Defined Driver Programs in RC3600
- [3] RCSL No.: 31-D477, January 1978, Tove Ann Aris and
Bo Tveden-Jørgensen
RC8000 Monitor, Part 2, Reference Manual
- [4] RCSL No.: 31-D455, June 1978, Palle Andersson
Operating Systems, Reference Manual
- [5] RCSL No.: 43-Ri0635, December 1976, Erik Lilholt
RCNET, General Information
- [6] RCSL No.: 31-D421, February 1977, Lars Otto Kjær Nielsen
and Bent Bæk Jensen
BOSS2 Installation and Maintenance
- [7] RCSL No.: 43-Ri0611, June 1977, Erik Lilholt
RCNET, RC3600 General Host Interface
- [8] RCSL No.: 31-D555, Under preparation
RC8000 Host Process

B. PROTECTION SYSTEM FOR PROCESSES AND CATALOG ENTRIES.

To each process and catalog entry are associated a name and a base (process base/entry base) consisting of 2 integers: lowerbase and upperbase. The process/entry base defines from where the process/entry is visible relative to internal processes.

Associated with each internal process are furthermore two protection bases, the maxbase and the stdbase, defining the access rights of the process, and a viewpoint base, the catbase, from which it looks for visible processes/catalog entries.

The process base, maxbase, stdbase, and catalog base are part of the process description in the monitor and are initialized by the parent, when the internal process is created.

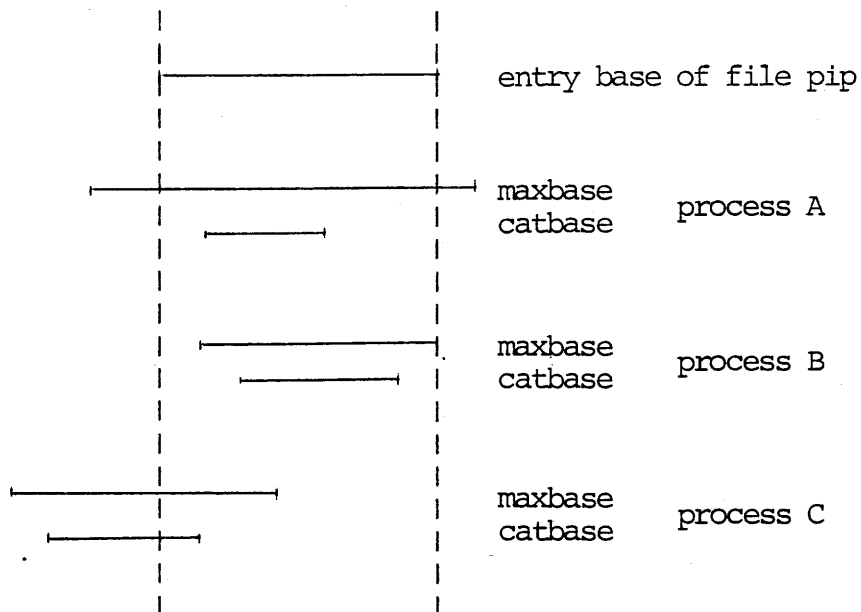
The catbase may freely be changed by the internal process as long as catbase is inside stdbase or between stdbase and maxbase.

The read and write access rights to files (catalog entries and their associated data areas) are defined by the following rules:

- An internal process may read a file if its catbase is inside or equals the entry base of the file (the file is visible to the internal process).
- An internal process may modify or remove a file, if the file is visible to the process and its maxbase is surrounding or equals the entry base of the file (the file is not protected against the internal process).
- When more than one file with the same name is visible, the one with the smallest entry base surrounding the catbase of the internal process is selected.

Analogous rules are valid for the communication between an internal process and another process (e.g. a link process).

Examples:



The file pip is visible to process A and B but not to process C and process A may write in the file.

Associated with each catalog entry is a permkey (catalog key), which is used by the operating systems to determine the "life time" of the entry.

Permkey together with the entry base represents the cocalled scope of an entry and relative to an internal process the scope is defined in the following way:

temp	permkey 0 or 1, entry base equals (or is contained in) the stdbase.
login	permkey 2, entry base as for scope temp.
user	permkey 3, entry base equals user base (explained below).
project	permkey 3, entry base equals maxbase.
system	permkey 3, entry base surrounds maxbase.

When a catalog entry is created, its permkey is initialized to 0 and the entry base is set to catbase of the calling process, indicating a temporary entry.

The scope of an entry can be changed by the utility program "scope" or BOSS command "scope" or by means of the monitor procedures "permanent entry" and "set entry base".

When the operating system s creates a child process maxbase, stdbase, and catbase of the process are assigned the equivalent values from the console parameters (maxbase, stdbase, userbase). The base of the process name is set to the max interval (-8388607, 8388606).

In case of a job command, the console parameters are taken from the s-user-catalog.

In case of a new command the console parameters maxbase, stdbase, and userbase are taken.

In case of an all command the console parameters are set to the max interval.

The console parameters maxbase, stdbase, and userbase can be changed by means of the s-commands base, project, login, and user. The default values are (8388606, 8388606).

When the process is started the file processor initializes its two variables, called userbase, to catbase of the process and catbase is set to stdbase.

When the operating system BOSS creates a child process maxbase and catbase are assigned the equivalent values from the BOSS-user-catalog (maxbase and userbase) and stdbase is calculated as a function of userbase (from the user-catalog) and the user index (from the login procedure), so that userbase surrounds the stdbase. When fp is initialized userbase (in fp) is set to catbase and catbase is set to stdbase.

An example of the base allocation for a BOSS project may look like:

