RCSL No: 991 09916

Edition: October 1984

Author: Finn G. Strøbech

Title:

RC8000 Utility Program

Disccopy.

Packon, Packoff.

Kiton, Kitoff, Kitlabel, Kitname.

RCSL Nr. 46-F 0087

A/S REGNECENTRALEN
af 1979

Keywords:

RC8000, RC4000, disc, backup.

Abstract:

The disccopy program is intended for binary copying between discs and for save and load of backing storage files using discs as backup medium.
The programs packon and packoff include and exclude entire RC83xx discpacks and their logical discs into and from the backing storage system.
The programs kiton and kitoff include and exclude single logical discs into and from the backing storage system.
The program kitlabel labels a single logical disc while the program kitname just writes a name into the label.
(26 printed pages)

Printed by A/S Regnecentralen af 1979, Copenhagen

RCSL Nr. 46-F 0087

TABLE OF CONTENTS PAGE

1. INTRODUCTION.

The program disccopy is primarily intended for:

- binary copying of discpacks or parts of discpacks from one discpack to another.

- save and load of backing storage files, using discpacks as backup medium.

In addition to this the program contains facilities, which make it possible to:

- include and exclude discs into and from the backing storage system.

- label a disc.

- rename a disc.

The program can be used in two different modes:

- as an ordinary utility program.

- as a conversational program, which resides permantly ('corelocked') in core and reads its parameters from the current input zone. This gives you complete freedom to copy discs at installations with only two discdrives.

Following subprograms are contained in the program disccopy:

save: used to save backing storage files on a disc.

load: used to load backing storage files from a disc.

bin: used for binary copying from one disc to another.

packon: includes an entire RC83xx disc in the backing storage system.

packoff: excludes an entire RC83xx disc from the backing storage system.

kiton: includes a single disc in the backing storage system.

kitoff: excludes a single disc from the backing storage system.

kitlabel: used to label a disc, i.e. names are assigned to 'document' and 'auxiliary catalog' and an empty auxiliary catalog is written.

kitname: assigns new names to 'document' and 'auxiliary catalog' of a disc (to be explained later).

The program must run in an 'all' process or in a 'new'process with a function mask, which allows the process to call the privileged monitor procedures for:

- auxiliary catalog handling.
- main catalog handling.
- create peripheral process.
- remove peripheral process.
- auxiliary entry handling.

When executing in the conversational mode, the program demands a process size of at least 80000 halfwords. Executing as normal utility program, the program demands a minimum size of 2000 halfwords.

2. CALL.

2.1 Utility Program Mode.

When the program disccopy is used as a normal utility program, the call syntax is as follows:

```
disccopy  { save   <savespec>
          { load   <loadspec>
          { bin    { <binary params> }1 0
```

```
<savespec>::= <saveparams> { list.{yes}1 0 { <criteria> }1 0
                           {no}
                 { <names> }1 0
```

```
<loadspec>::= <loadparams> { list.{yes}1 0 { <criteria> }1 0
                           {no}
                 { <names> }1 0
```

```
<saveparams>::= { to. <devno>                  }2 2
                { { from. <docname> }1 0       }
```

```
<loadparams>::= { to. <docname>                }2 2
                { from. <docname>              }
```

```
                      scope. { system
                             { project
                             { user
<criteria>::=
                      base.  { system
                             { project
                             { user
                             { <lower>.<upper>
```

```
<names>::= { <name> }* 0
```

<binary params>::= [to. <devno> {.all}(0,1) | from. <devno> {.all}(0,1)](1,2)

The programs packon, packoff, kiton, kitoff, kitlabel and kitname are called as utility programs as follows:

packon devno.<devno> [list. {yes | no | nonsys | error | warning}](0,1)

packoff devno.<devno>

kiton devno.<devno> [list. {yes | no | nonsys | error | warning}](0,1)

kitoff devno.<devno>

kitlabel <devno> <docname> <auxname> {slow | fast}(1,1) <catsize>

<slicelength> <size>

kitname <devno> <docname> <auxname>

<devno>, <lower>, <upper>, <catsize>, <slicelength>, <size> ::= integer

<name>, <docname>, <auxname> ::= name, max. 11 chars.

2.2 Conversational Mode.

Conversational mode is entered by the call:

```
disccopy typein
```

Now the subprograms are activated as follows:

```
save <savespec>
load <loadspec>
bin
```

The programs packon, packoff, kiton, kitoff, kitlabel, and kitname are activated as in utility program mode, i.e.

```
packon devno.<devno>    list.{ yes     }1
                             { no      }
                             { nonsys  }
                             { warning }
                             { error   }0
```

```
etc.
```

When all your tasks are completed, you leave the conversational mode by typing the command:

```
end
```

3. DETAILS.

In this section details are given about the subprograms and their parameters.

3.1 Save.

This subprogram will select backing storage files as specified by the call parameters <criteria> and <names> and save these entries and their corresponding areas (if any) on the object disc.

Parameters:

to.<devno>

The parameter specifies the object disc.
Files are copied to the disc with devicenumber <devno>.
The object disc must not be included in the backing storage system and the files which are copied must not exist in the auxiliary catalog of the object disc (no copying is performed if this is the case).

from.<docname>

The parameter specifies the source disc.
This parameter is optional. If source disc is specified, it must be included in the backing storage system and only files which belong to this disc are copied. If no source disc is specified, all files which satisfy the conditions stated by <criteria> and <names> are copied.

<criteria>

scope: only files with the scope specified are saved.

base: only files with entrybases inside the interval specified are saved (base.system is equivalent to base.-8388607.8388605).

Negative basevalues can only be read in conversational mode, as fp does not accept negative integers, e.g.
save to.7 base.-8388607.8388605 is valid only in conversational mode, while
save to.7 base.410.419 is valid in both utility program and conversational mode.

If no <criteria> are specified, default is scope.system.

<names>

A list of entrynames can be used to reduce the <criteria> parameter, i.e. only files with name and scope/base as specified are saved.

list. {yes / no}

A listning of the files saved is produced on current output if list.yes is specified. list.no is default.

Examples:

Ex.1. All files which belong to 'disc' and have user-scope are saved on the disc mounted on device 7 (but not included in the backing storage system) by the following call:
save from.disc to.7 scope.user list.yes
A listning of the files saved is produced on current output.

Ex.2. By the following call, all backing storage files which fulfil:
-8388607<= lower entrybase <= upper entrybase <= 8388605 are saved on the disc, which is mounted on device 7:
save to.7 base.system

Ex.3. By the following call, the two backing storage files ncjfile1 and ncjfile2, which have system scope (default) are saved on the disc, mounted on device 7:

```
save to.7 ncjfile1  ncjfile2
```

3.2 Load.

This subprogram will select files on the source disc as specified by <criteria> and <names>. The selected files are loaded into the backing storage system, i.e. non existing entries are created on object disc and existing files on object disc are overwritten.

If a file is already included from a document different from object disc, load of the file is rejected and a message is written on current output. In this case it will be necessary to rename the entry which causes conflict and repeat the load.

Parameters:

to. <docname>

Specifies the object disc.
The object disc must be included in the backing storage system.

from. <devno>

Specifies the source disc.
The source disc must not be included in the backing storage system.

<criteria>, <names> and list

As for save.

Example:

All files of scope user on the disc, which is mounted on device 6 (but not included in the backing storage system) are loaded to 'discl' (which must be included in the backing storage system) by the following command:

```
load from.6 to.discl scope.user
```

3.3 Bin.

This subprogram is used for various kinds of binary copying between discs. The following remarks concern binary copying in general:

- For safety reasons you should writeprotect the source disc (not possible for some fixed media discs and diablodiscs).

- The object disc must not be included in the backing storage system. You must explicitly perform kitoff on the object disc if this is the case.

- kitoff is implicitly performed on the source disc. During this possibly the text:

  notice: disc with maincatalog is removed

  will appear. If this is the case, it will be necessary to perform one of the following actions (depending on the "programmode") in order to connect the maincatalog when the copying is terminated:

utility mode:	switch off the writeprotection and autoload the system.
conversational mode:	switch off the writeprotection and perform kiton on disc with main catalog.

The same procedure must be used if the object disc during the copying has replaced the disc containing the main catalog. Remember to reinstall the main catalog disc first.

- When the copying is terminated, the text:

  copying terminated
  number of segments copied: <segments>

  is written on current output.

3.3.1 Bin with <binary params>.

When the subprogram is called with <binary params>, it is used for copying from one disc to another, as described below:

Parameters:

from/to. <devno> .all

'all'-parameter present: copying is performed from/to the physical disc, which contains the logical disc with devicenumber <devno>.

no 'all'-parameter: copying is performed from/to the disc (physical or logical) with devicenumber <devno>.

Examples:

```
bin to.7 from.6          (logical disc -> logical disc)

bin to.7.all from.6.all  (physical disc -> physical disc)

bin to.7 from.6.all      (physical disc -> logical disc)
```

3.3.2 bin without parameters.

When called without parameters, the subprogram bin is used for copying of specified parts of discs or for copying between discs of unequal size (examples of this are given below). If the subprogram bin is called without parameters, it will ask for:

```
to device:
start segment:
from device:
start segment:
number of segments:
```

To every question, you must type in a number, terminated by carriage return (<devicenumber> respectively <segmentnumber>). In case you do not know the size of the disc, just type in a too big integer.

Example 1:

In this example a 33 Mb disc is copied from device 6 to device 7.

```
to device:          7
start segment:      0
from device:        6
start segment:      0
number of segments: 43155
```

Example 2:

In this example is shown how to copy a 66 Mb disc on two 33 Mb discs.

```
to device:          6    (33 Mb)
start segment:      0
from device:        7    (66 Mb)
start segment:      0
number of segments: 43155
```

```
to device:          6
start segment:     0
from device:       7
start segment:     43155
number of segments: 43155
```

It is impossible to copy the entire disc, but all slices containing data are copied (the last 105 segments are not copied).

In the conversational mode, the subprogram must be explicitly activated for each copying by typing the text:

```
bin
```

3.3.3 Sizes of Disc and Copytime.

size in Mb	size in segments	copytime in minutes
2,4	2424	1/2
12	15435	3/4
33	43155	1 1/2
66	86415	3
66	87146	3
124	163989	6
133	174293	6 1/2
248	328377	12

3.4 Packon.

The program is used to include into the backing storage system an entire RC83xx disc, i.e. all the logical discs it contains.
The remarks below for kiton are valid for each disc included by packon.

Parameters:

devno. <devno>

The device specified must be the device number of the physical disc to be included.

list

As for kiton.

3.5 Packoff.

The program is used to exclude from the backing storage system an entire RC83xx disc, i.e. all the logical discs it contains.
The remarks below for kitoff ar valid for each disc excluded by packoff.

Parameters:

devno. <devno>

The device specified must be the device number of the physical disc to be excluded.

3.6 Kiton.

The program is used to include a logical disc in the backing storage system. If the maincatalog has been removed and the auxiliary catalog of the disc involved contains an entry with the name <:catalog:>, this entry is selected as maincatalog and the maincatalog is connected.
This is very convenient, as you need not autoload the system if the maincatalog has been removed and you want to switch to a task which demands the presence of the maincatalog (when in the conversational mode).

Parameters:

devno. <devno>

The logical disc with devicenumber <devno> is included in the backing storage system.

list

yes: all entries in the auxiliary catalog are listed on current output during insertion in the main-catalog. If an insertion caused trouble, the monitor result is listed, too.

no: no listing.

nonsys: listing of all non system entries, which have been inserted in the main catalog.

warning: listing of entries, which have not been inserted due to nameoverlap.

error: listing of entries, which caused trouble during insertion. The monitor result is listed too.

Example:

kiton devno.7 list.yes

3.7 Kitoff.

The program excludes a logical disc from the backing storage system. If the disc involved contains the main-catalog, the maincatalog is removed and the text:

notice: disc with maincatalog is removed

is written on current output.

If the program itself resides on the disc involved, the text:

notice: disc with program file is removed

is written on current output. This causes no trouble if you run in the conversational mode. If in normal utility mode, the program must be reinstalled before further use.

If the file processor is found on the disc, the text:

notice: disc with fp is removed

is written on current output. In conversational mode it represents no problem until the end command is given.

When the program terminates, it sends a finis message to the parent.

Parameters:

devno. <devno>

The logical disc with devicenumber <devno> is excluded from the backing storage system.

Example:

kitoff devno.6

3.8 Kitlabel.

The program assigns names to 'document name' and name of 'auxiliary catalog' of a logical disc and writes an empty auxiliary catalog on the disc.

Parameters:

<devno>	: devicenumber
<docname>	: documentname
<auxname>	: name of auxiliary catalog
slow/fast	: disc/drum, respectively
<catsize>	: size of auxiliary catalog in segments
<slicelength>	: segments/slice
<size>	: discsize in slices

Example:

```
kitlabel 7  discl  catdiscl  slow 50 21 2045
```

3.9 Kitname.

The program is introduced as an 'ad hoc' solution to a problem, which arises if the program disccopy terminates during a save. During a save, the object disc is included in the backing storage system with workingnames for "document" and "auxiliary catalog" in order to avoid confusion if nameoverlap occurs. These workingnames are dumped in the chainhead of the object disc each time an entry is created on the object disc. Normally, this causes no trouble as the original names are rewritten when the program terminates properly. If, however, the program terminates during the save, you will have to use the program kitname if you want to reestablish the original names without erasing the files saved on the disc.

Example:

```
kitname 7  discl  catdiscl
```

4. ERRORMESSAGES AND WARNING.

4.1 Logical Status.

If the program disccopy finds it impossible to read or write a segment, one of the following errormessages are written:

input from <name> segm: <segnumber> status <bitpattern>
output to <name> segm: <segnumber> status <bitpattern>

where <bitpattern> is a logical statusword, describing the read/write-operation, cf. ref. (1), chapter 6.

The copying will ignore the segment and continue, through, with fp mode bits warning.yes and ok.no, with following exceptions:

a. <bitpattern> = ...1.....1.
b. <bitpattern> = ...1.....

a. means end of document, i.e. the object disc is too small.
b. means receiver does not exist, i.e. the discdrive is switched off, and the like.

4.2 Results from the Monitor.

The program responds to errors returned from monitor-calls by writing a text describing the monitor procedure and the actual result value, e.g.

create peripheral process <name> result <number>

When executing in utility program mode, the program will terminate and return to fp with modebits ok.no, warning.yes.

When executing in conversational mode, the program will continue reading commands whitout touching the mode bits.

The list below describes all errormessages of this type. In parenthesis the numbers of the corresponding monitorprocedures are given.

create peripheral process	(54)
delete bs	(108)
delete entries	(110)
prepare bs	(102)
set catalog base	(72)
create entry	(40)
create aux entry	(120)
insert entry	(104)
insert bs	(106)

When one of these errors occurs, please consult ref. (2) to get the details.

4.3 Warnings.

When running in utility program mode, the program will return to FP after normal termination with modebits ok.yes, warning.yes in case of warnings.

4.3.1 Save.

If you attempt to save an entry which already exists in the auxiliary catalog of the object disc the following warning is written:

entry already exists in auxcat: <entryname>

and the entry is skipped.

4.3.2 Load.

If an entry to be loaded is included in the backing storage system from a document different from the object disc, the following warning is written:

entry already included from another document:
<entryname>

and the entry is skipped.

4.3.3 Both.

If one or more entries specified cannot be found in the catalog/auxiliary catalog, the warning

*** entries not found

succeded by a list of the entries concerned is written

5. REFERENCES.

(1) RCSL No 31-D676: System 3 Utility Program, Part one.

(2) RCSL No 31-D617: RC8000 Monitor, Part 2, Reference Manual.

(3) RCSL No 31-D643: Operating System s, Reference Manual.

RETURN LETTER

Title: DISCCOPY, PACKON, PACKOFF, KITON, KITOFF, KITLABEL, KITNAME. RCSL No.: 999 09916

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

RCSL Nr. 46-F 0093

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

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

Affix
postage
here

REGNECENTRALEN
af 1979

Information Department
Lautrupbjerg 1
DK-2750 Ballerup
Denmark