
RCSL No: 31-D712

Edition: September 1983

Author: Carl Henrik Dreyer

Title:

RC8000 ASYNCHRONOUS COMMUNICATION PACKAGE (ACP8000)
Installation Guide/Operating Guide.

Keywords:

RC8000, File Transfer, Asynchronous Terminals, Installation Guide, Operating Guide, Basic Software, AMX, CP/M.

Abstract:

ACP8000 is used for file transfer between RC8000 and intelligent terminals, connected via an Asynchronous Multiplexer (AMX). This manual describes the installation and System trimming procedure, operating instructions, and some relevant details on the conventions used.

(40 printed pages)

Copyright © 1983, 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.

FORWORD

First edition: RCSL no 31-D712.

This manual is included in the first general release of the RC8000 Software package:

SW8713: RC8000 Asynchronous Communication Package (ACP8000).

TABLE OF CONTENTS	PAGE
1. INTRODUCTION	1
2. INSTALLATION GUIDE	2
2.1 User Catalog	2
2.2 System Trimming	5
2.3 Process Requirements	7
3. OPERATING GUIDE	9
3.1 Start up	9
3.2 Normal Operation	9
3.3 Close Down	10
4. UTILITY PROGRAM NEWACPCAT	11
4.1 Examples	11
4.2 Call	11
4.3 Function	11
4.4 Parameters	12
4.5 Format of the Output	12
4.6 Error Messages	14
4.7 Warning Messages	15
5. IMPLEMENTATION DETAILS	16
5.1 Function Descriptions	16
5.2 Command Syntax	19
5.3 Filename and Mask Syntax	20
5.4 Catalog Entry Handling	22
5.5 Text File Conversion	23
 <u>APPENDICES:</u>	
A. REFERENCES	25
B. USE OF TEXTS IN ACPTRIM	26
C. START UP ALARMS FROM BACP	29
D. PROCEDURE WRITESTATUS	30

1. INTRODUCTION

1.

The ACP8000 system is used for file transfer between RC8000 and an intelligent terminal, connected to an RC8000 device host by an AMX (asynchronous multiplexer). The intelligent terminal may be an RC855 or RC700, running the corresponding ACP program.

As to the ACP8000 system, the only demand is that the terminal obeys the protocol used.

A general introduction to the system is found in ref. [1].

Chapter 2 describes the installation process of ACP8000, including user catalog, disc resources, system parameters and system trimming.

Chapter 3, contains operating guide, i.e. normal startup and closedown.

Chapter 4 describes the utility program newacpcat, used in maintaining the user catalog.

Chapter 5 contains interesting details which may be relevant for both users and operators.

In this manual, all character values stated refer to the ISO alfabeth, and the values are in plain decimal representation.

If you are very keen in getting started, the following will do:

- read section 2.1, and make a small user catalog, skip corrections in trimming (section 2.2) and start the system with the standard startup file sacp.

2. INSTALLATION GUIDE

2.

The first steps of installation (loading programs etc.) is described in the Package Description. This chapter describes what to do to get a proper running system established. 3 tasks must be performed:

1. Establish a user catalog, using the utility program newacpcat described in chapter 4, reading conventions in section 2.1 below.
2. Consider system parameters and texts, correct the file acptrim, and compile the trimmed system by ALGOL. Follow section 2.2 below.
3. Consider the process startup parameters by correcting the "read-and-start" file sacp. Follow section 2.3 below.

The first two steps should be performed in a process with bases equal to max base of ACP process (surrounding all user bases, cf. section 2.1). This base need not be system base. Step 3 should be done at system base, where the file sacp must be placed.

2.1 User Catalog

2.1

The user catalog is a disc file containing descriptions of all users. The file is a hash-file, the hash key being (user name, number). The hash function gives a segment number (home segment). Overflow is placed in the file itself - in first free position (scanning cyclic). The capacity is 16 records per segment, as one record uses 32 halfwords.

The key (user name, number) must be unique. The catalog has the same format as the ACCESS DOOR CATALOG in RC8000 SHIPPING, cf. ref. [3]. Several terminals may be logged in simultaneously, using the same (user name, number).

Note that the user catalog cannot be changed during ACP8000 operation. This must be done in advance.

The user catalog is established using the utility program newacpcat (cf. chapter 4).

Contents of a user catalog record:

- User name:
A name of max 11 characters. Part of the key.
- User number:
An integer number (may be negative). Part of the key. If not stated, zero is assumed.
- Lower base, upper base:
Two integers, specifying the catalog base where to find the "private" files of the user. The user can only obtain write access to files having this base, and a newly created file will get this base. Read access is obtained as "seen from" this base, but also controlled by system parameter visibility (cf. section 2.2).
- Access mode:
A bit mask giving the access rights of the user. See below.
- File name:
A name of max. 11 characters. If specified, the user can only access a file of this name. If not specified, no restriction is invoked.
- Insert time:
A shortclock, the time of insert (by utility program newacpcat).
- Last use time:
A shortclock, time of last login.

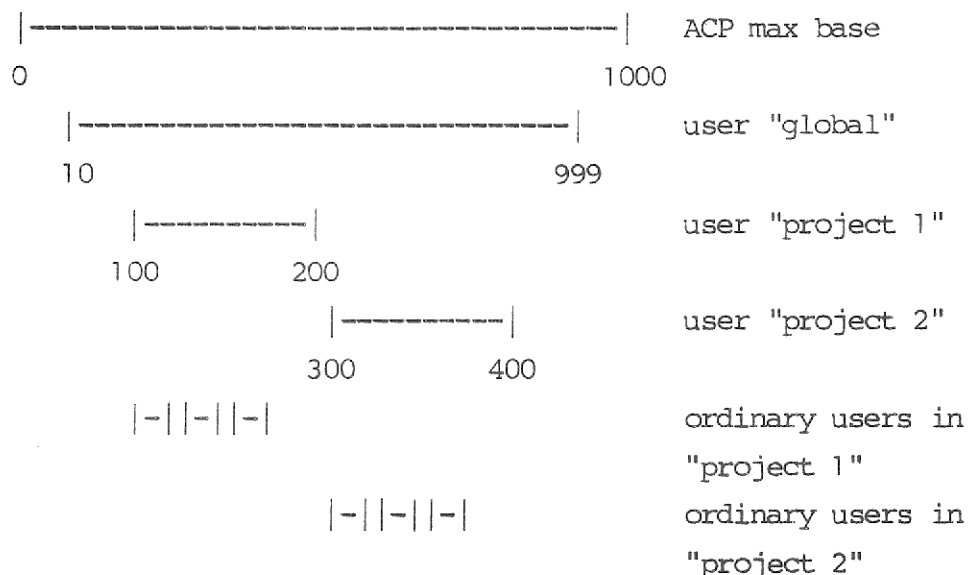
Access mode:

Each bit permits (if stated) the function mentioned. Mode char below is the character used by newacpcat for describing the bit. The functions are described in section 5.1.

Mode char = Name	Description
o = onewrite	write access is permitted once, i.e: after first successfull transfer, this bit and the write bit is cleared (and functions 1,3). The bit allows functions 1, 3
p = print	print command, i.e. function 8.
l = lookup	permits survey command, i.e. function 4.
r = read	read access allowed, i.e. functions 0, 2.
w = write	write access allowed, i.e. functions 1, 3. Furthermore this bit permits functions 5 (clear) and 7 (close).

Base structure of user catalog:

Each user has a base: this is an interval, used for catalog lookup on files. Files having the user base are private to the user. The base may be used for constructing a hierarchy of users, for example:



The ordinary users may read files on "user global" base, and the corresponding project base, but not write them. They cannot read other user's files.

The project users can read files on "user global" base (and their own files), but not any user files.

Note that no user base may be outside process max base of the ACP process.

2.2 System Trimming

2.2

The system contains an ALGOL program text: acptrim. The trimming is done by editing this text, and compile it with the ALGOL compiler. Of course you may skip the trimming if you are satisfied with the standard.

The program text contains:

1. procedure writestatus: used for writing final result of a transfer. Called after each normal transfer of one file. Also called after transfer of a directory, if the transfer was not ok. Details in App. D.

2. Initialization of system parameters:

maxterm: Maximum number of terminal users, that can be logged in simultaneously. The value has no relevans to the physical number of terminals connected to RC8000, nor the number of users in the user catalog.

par(1)-par(4): Name of file holding user catalog. Must be present at process max bases.

par(5)-par(8): Name of workfile. If present, it must be at process max bases. If not present, it is created with scope temp. The size is always set/extended to:

$(\text{maxterm}+1)*\text{par}(9)$ segments.

The file is used for holding directories.

- par(9): Work-segments per terminal. Gives an upper limit to directories, as each segment can hold 64 directory names.
- par(10): Number of segments in shared buffer, used for scanning RC8000 main catalog. A small buffer gives lower primary storage requirements, but slower maincat scan.
- par(11): Slicelength: number of segments used when a file is created/extended, during write. The most efficient value is the slicelength used on the disc where the process is given permanent resources.
- par(12): Visibility: controls the visibility of files with a base outside the user base:
 1 = only files with user base visible.
 2 = all files (visible from user base) with base inside or equal to process max base are visible.
 3 = all files (visible from user base) are visible, i.e. also system files.

Note that this parameter is only used when reading a file specified by name. If a file is specified by a mask, only files having user base are examined.

- par(13)-(15): Document name of disc, where new files are created.
3. initialization of texts (for terminal dialog):
 long array txt (1:34). All texts may be changed. Cf. App. B.
 4. Initialization of error texts (for terminal dialog):
 long array err (1:12). All texts may be changed. Cf. App. B.
 5. Initialization of command texts:
 long array com (1:14). The array contains short text strings, appended with corresponding function number. The texts cannot be more than 5 characters long. Several texts may correspond

to the same function. If more texts are identical, only the first one will be recognized. The array may be extended as much as wanted, but all elements should be assigned.

6. Call of procedure acpproc.

This procedure contains the actual system, and the call need not be altered.

Example of system trimming:

The following commands changes the system parameters maxterm and visibility:

```
chdtrim = edit acptrim
1./maxterm:=/, r/6/8/
1./par(12):=/, r/3/2/
f
bacp = algol chdtrim
scope user bacp.
```

2.3 Process Requirements

2.3

The ACP process is running at the RC8000 under the basic operating system "s". All process parameters are set in the file sacp. This file is delivered with a default content, and should be edited to match the actual system. Besides, the file is used for primary input for the process. This makes the system startup simpler.

The file sacp must have system scope, as it is used for a read-command to s.

Process bases: all bases of the process must equal process max base, which should surround all bases in the user catalog.

Disc ressources: The process should have disc ressources for creating new files. Note that all disc ressources available for the process are shared between the users. There is no ressource restrictions on the single user. This implies, that one user may

take up all disc resources for his own private files. Note that the disc resources given to the process are only spent when new files are created, as existing files are not counted here. And the resources are increased when files are cleared. Section 5.4 contains details on file creation and deletion. Give the process permanent resources on the disc stated in the system trimming: par(13)-par(15).

Runtime files: At process start 2 files are needed, at process maxbase:

- program file: bacp
- user catalog: acpcat

A third file (the work file: acpwrk) will be created if it is not present at max base.

All 3 files will automatically be protected against both reading, writing and clearing by any user, so no specific protection is needed.

Example of read-and-start file sacp:

```
*new acp
*base 0 8388605      ; all bases equal.
                    ; the files bacp and acpcat should be at this
                    ; scope.
*size 36000         ; 24000 + 2000 x maxterm
*buf 15             ; 9 + maxterm
*area 13            ; 7 + maxterm
*perm disc 0 0      ; no resources at disc
*perm disc1 20 500 ; disc resources for new files
* i 4.sacp          ; this file = primary input
* run unstack       ; start the process
bacp                ; call of program
finis               ; finis process
```

3. OPERATING GUIDE

3.

3.1 Start up

3.1

The system is started by the command to s:

```
att s
read sacp
```

After some seconds the system writes

```
cat is acpcat segm = 20 d.830902.1455 users = 32 = 10%
acp8000 rel.1.0 started, max term = 6 paging hws = 9872
```

or some alarm occurs, cf. appendix C.

The startup should be done at a console with paper, in order to keep a hard copy of eventual errors.

If the user catalog is to be changed, this must be done before startup.

3.2 Normal Operation

3.2

During operation, no operator actions are needed.

You may experience ALGOL alarm output, in case program errors occur. But usually the system is going on operating, only logging out the affected user.

No normal output appears on the startup console.

All output for console is actually written on current output.

The normal close:

Log in to the system and give the close command:

```
att acp
acp8000 1.0
type userid: operator
function (h for help): close
closed by operator
```

```
end of acp, logins = 27 moves = 38 bytes moved = 213715
cpu = 363 secs
from s
pause acp finis fp
```

The brutal close:

```
att s
proc acp break
```

I will strongly warn you against using the "remove" command to get rid of the process. If some users are in the middle of a transfer, the AMX port in question will be useless for terminal connection until next autoload.

4. UTILITY PROGRAM NEWACPCAT

4.

This program is intended for maintenance of the ACP user catalog: initialization of a new catalog, and reorganization of an existing one.

4.1 Examples

4.1

```
acpcat = newacpcat
```

will initialize a new empty catalog in the (existing) file acpcat.

```
acpcat = newacpcat cat1
```

will perform a reorganization from cat1 into acpcat.

4.2 Call

4.2

$$(\langle \text{newcat} \rangle =) \begin{matrix} 1 \\ 0 \end{matrix} \text{newacpcat} \left\{ \begin{matrix} \langle \text{oldcat} \rangle \\ \text{init.yes} \end{matrix} \right\} \begin{matrix} 1 \\ 0 \end{matrix} \left\{ \begin{matrix} \text{list.yes} \\ \text{list.no} \end{matrix} \right\} \begin{matrix} 1 \\ 0 \end{matrix}$$

4.3 Function

4.3

1. if $\langle \text{newcat} \rangle$ is specified, a new (empty) catalog is initialized.
2. if both $\langle \text{newcat} \rangle$ and $\langle \text{oldcat} \rangle$ is specified, all used records in $\langle \text{oldcat} \rangle$ are inserted in $\langle \text{newcat} \rangle$.
3. if `init.yes` is specified, user records are inserted in $\langle \text{newcat} \rangle$, reading data (door contents) from current input.
4. write a survey of $\langle \text{newcat} \rangle$ (if specified, else $\langle \text{oldcat} \rangle$, if specified).

5. if list.yes is specified, a survey of all users in <newcat> is output (if <newcat> specified, else <oldcat>, if specified). The output has a form that is readable for newacpcat in mode init.yes.

4.4 Parameters

4.4

- <newcat> File names. The files must exist in advance. They
<oldcat> need not be of same size.
- list.yes A listing is wanted. Default is list.no.
- init.yes The contents of user records to insert is read from current input, terminating on the word 'end' (as a user name).

4.5 Format of the Output

4.5

The output at list.yes contains one line per user, with the information from the user entry in a form readable for newacpcat. After the last-use-time is output information about the actual location in the file.

The users are not sorted, but output as they are found by scanning the file. On the next page, an example is shown.

***** job: *****

mode list.yes
acpcat=set 84
acpcat=newacpcat init.yes list.yes
rcdemo 51 900 939 plrw,
usera 51 900 909 lr,
leader 51 900 909 lrw,
useronce 51 910 919 o,
onefile 51 910 919 lw,ufile
oscar 51 920 929 or,ofile
classroom 0 1000 1099,
teacher 0 1000 1099 plwr,
student 1 1000 1009 lr,
student 2 1010 1019 lr,
teacher 1 1000 1009 pw,
end

***** job output: *****
*acpcat=set 84
*acpcat=newacpcat init.yes list.yes

acpcat initialized with 11 doors, from 12 lines
survey of acpcat segm 84 dated 831007.1057
; capacity 1344 100 overflows no pct
; used 11 1 max overl 0 0
; deleted 0 0 max utiliz 1 6
; free 1333 99 full segm 0 0

name	no	lower	upper	access	file	inserted	used
teacher	0	1000	1099	pl rw	;831007.1057;831007.1057;	6.0	6
teacher	1	1000	1009	p w	;831007.1057;831007.1057;	7.0	7
useronce	51	910	919	o	;831007.1057;831007.1057;	10.0	10
classroom	0	1000	1099		;831007.1057;831007.1057;	20.0	20
usera	51	900	909	l r	;831007.1057;831007.1057;	25.0	25
rcdemo	51	900	939	pl rw	;831007.1057;831007.1057;	27.0	27
oscar	51	920	929	o r,ofile	;831007.1057;831007.1057;	28.0	28
leader	51	900	909	l rw	;831007.1057;831007.1057;	30.0	30
onefile	51	910	919	l w,ufile	;831007.1057;831007.1057;	49.0	49
student	1	1000	1009	l r	;831007.1057;831007.1057;	65.0	65
student	2	1010	1019	l r	;831007.1057;831007.1057;	66.0	66

Note that the "end" will terminate a later reading.

Note that the semicolons make the rest of the line a comment.

All output is written on current output medium.

4.6 Error Messages

4.6

Any of these error message will set FP mode to ok.no.

```
xxxnewacpcat sorry <file> not found
    <file> is either <oldcat> or <newcat>
```

```
xxxnewacpcat sorry, no left hand side
    when init.yes is specified, a lefthand side must be present.
```

```
xxxnewacpcat sorry, unknown parameter <param>
    the unknown parameter is written.
```

```
xxxnewacpcat sorry, same file.
    <newcat> and <oldcat> are the same.
```

```
xxxnewacpcat sorry <file> not a acp usercat
    detected when the survey should be made. Work 10 of entry tail
    <> no of segments.
```

When reading from current input, the following error messages may occur, all headed by

```
xxxnewacpcat sorry, line <line>:
    where <line> is current line number read.
```

```
The rest of the line may be
    line too long
        program buffer too small
    line not complete
        missing information
```

illegal character

only small letters, digits allowed

syntax at key no

key name has been read

syntax at scope

syntax at access mode

illegal access mode

line termination

too much information

syntax at key name

not a name, or name too long.

syntax at file name

not a name, or name too long.

4.7 Warning Messages

4.7

The message

key conflict <name> <no> scope <low> <up> line <line>

is written when two user keys (name, number) are the same. In the init.yes-function, the line number is output (i.e. line no of last occurrence). The FP mode is set warning.yes, but the program continues.

The message

file full: <file>, <users> inserted

indicates a too small <file>. The function (either "reorg" or "init.yes") is left, and the FP mode is set warning.yes.

The message

newacpcat: warning

is written at program termination, if the FP mode was set warning.yes.

5. IMPLEMENTATION DETAILS

5.

This chapter describes some details of the system, that may affect both operator and users. Appendix B gives a complete list of texts used by the system.

5.1 Function Descriptions

5.1

The functions are selected via the array com (cf. section 2.2).

Login

When receiving an attention message from a process not already logged in (this is actually what is performed when the ESC-button is hit at the terminal, "att" is written and the name of the ACP process is typed), the user name and number must be specified (number is 0 if not specified).

If the user is found in the user catalog, and the user bases are accessible (inside ACP max base), login is accepted, and user record updated (last use = time).

The access mode is used to check which functions are allowed, and file name is read.

Note that all output to the terminal is written in mode 2.

Command reading

After login, and after execution of any function, all previously typed input is dropped, txt(2) is output, and a command line is awaited from terminal keyboard. The first (max 5) text characters are compared to array com to find the function.

Function 0: send text

If a name (RC8000 file) is specified, a terminal file name should also be specified. The file is sent as text, terminating by either EM (ISO 25) or end of file.

If a mask is specified, the RC8000 main catalog is scanned to find all user files matching the mask. Then the text files are sent one by one, using exactly the same file name at the terminal and at RC8000.

If a mask is specified, a drive must also be specified. Then the RC8000 main catalog is scanned to find all user files matching the mask. Then the text files are sent one by one, using exactly the same file name at the terminal and at RC8000.

Function 1: receive text

If a name (terminal file) is specified, also an RC8000 file name should be specified. The file is received as text (cf. section 5.4 about file creation).

If a mask is specified, the terminal catalog is received, stating the mask. For every file name received from the terminal, the text file is received from the terminal, using the same file name at RC8000 and the terminal.

Function 2: Send binary

As function 0, except for

- no of bytes sent: if tail(9) = 0 and tail (10) > 0 then tail(10)
else <no of segments> * 768.
- all bytes are transferred as 8 bit bytes, without modification.

Function 3: receive binary

As function 1, except for

- all bytes are transferred as 8 bit bytes, without modification.

Function 4: survey

A drive is either not specified (= default terminal drive) or = txt(06) (=RC8000), or = a terminal drive.

A mask is also specified.

If drive = RC8000, the RC8000 main catalog is scanned, and all userfiles matching the mask are written. If some extra

parameter is specified after mask (and system trimming visibility >1), all files readable by the user (and with scope inside ACP max base) are taken.

If drive is not RC8000, the terminal catalog is received, stating the mask.

All file names received are written. The writing of file names resembles the CP/M command "dir".

Function 5: clear an RC8000 file

A file name must be specified.

The file must have user base, and be "not in use" (by anyone at RC8000).

The file is cleared, and the disc resources occupied by the file are available for the ACP process.

Function 6: logout

All connection with the terminal is dropped.

Function 7: close system

If any other terminal is logged in, the function is not performed.

If a number is specified as parameter, and this number is number of terminals logged in, the close is performed anyway.

Function 8: print

A file is specified. A printer may be specified (if not, default = txt (13) is used). The printer is actually a catalog entry describing a printer.

A "transfer" message is sent to the PRIMO system, and a result is written on the terminal.

If the print command is unsuccessful, a cause is printed. Possible values are:

2	=	rejected
4	=	receiver malfunction
5	=	receiver unknown
300	=	PRIMO no resources
501	=	troubles with the file
602	=	troubles with the printer

Function 10: unknown command

Command text not found in array com. txt (17) shown.

Function 11: empty command

No text typed before RETURN. No action done.

Function 12: help menu

txt(33) shown.

Function 13: terminal reservation

Eventual parameters in the command line are examined. If the letter "n" is found, the terminal process is reserved, otherwise it is released. After login the terminal is always reserved. Releasing the terminal enables the user to communicate with other processes during login, as ACP8000 after attention status during input awaits an attention message.

Note that the terminal is always reserved during file transfer. Note also, that output is not interrupted by attention status.

5.2 Command Syntax

5.2

$$(\langle \text{del} \rangle) \begin{matrix} 1 \\ 0 \end{matrix} \langle \text{command} \rangle (\langle \text{del} \rangle \langle \text{param} \rangle) \begin{matrix} * \\ 0 \end{matrix} \langle \text{term} \rangle$$

$\langle \text{command} \rangle$ is a text. The first 5 characters are used as command word, and searched for in array com.

$\langle \text{param} \rangle$ is a text. Any number of parameters may be typed, but the use depends on the function. Superfluous parameters are skipped. If a function needs a parameter which is not stated, a suitable text is written and parameter typing may continue.

$\langle \text{del} \rangle$ Any number (at least one) of delemitor characters, i.e. not text, blind or terminator. (See below).

<term> terminator character: NL (ISO 10) or EM (ISO 25), or CR (ISO 13), i.e.: use RETURN button.

text Both <command> and <param> are text strings, terminated by a delemitor or a terminator. Leading delemitors are skipped.

Characters included in text are:

- all letters (both small and capital)
- all digits
- the characters
 - * (ISO 42)
 - . (ISO 46)
 - : (ISO 58)

Blind are the characters NUL (ISO 0) and DEL (ISO 127).

5.3 Filename and Mask Syntax

5.3

A filename is one parameter, cf. section 5.2 (i.e. no spaces in between). The syntax is as close as possible to CP/M file names:

$$(\langle \text{drive} \rangle :) \begin{matrix} 1 \\ 0 \end{matrix} \langle \text{name} \rangle (. \langle \text{suffix} \rangle) \begin{matrix} 1 \\ 0 \end{matrix}$$

<drive> A one-character specification of drive (if several characters are specified, the last one is used). Usually a, b, c

If not stated, "default drive" is understood.

<name> min 1, max 11 characters (if <suffix> is stated, max 8). Superfluous characters are dropped. Small letters are converted to capital.

<suffix> min 1, max 3 characters. Small letters are converted to capital.

When a filename is written on the screen, capital letters are converted to small (except at catalog writing).

If <name> exceeds 8 characters and <suffix> is omitted, the last characters will represent <suffix>.

Mask: if a star (*) is met as a character in either <name> or <suffix>, it is not a file name, but a file mask. This will invoke a search for all file names matching the mask: the * means "any character", and if * is last character in <name> or <suffix> it means "the rest may be any character". If <suffix> is not stated, it is included in "any character". In short: it works as you would expect.

Mapping file names to RC8000 names:

<drive> is not used.

<name> is used as file name, converting capital to small letters. Note that the RC8000 file system demands a file name to have a letter as first character, and only small letters or digits in the rest. If no <suffix> is specified, this will be the file name.

<suffix> If specified, it is inserted as character 9, 10 and 11 (if 3 characters) of the name of total 11 characters. <name> is now extended to 8 characters by adding zeroes (ISO 48) if needed, constructing a 11 character name. Because of this, please observe the following

CAUTION: avoid using filenames ending with 0 (ISO 48) in <name> part.

Mapping file names to terminal file name.

All small letters are converted to capital. A name of 11 characters is constructed, filling unused positions with space (ISO 32).

Examples of file names (space shown as _):

<u>Typed</u>	<u>Terminal</u>	<u>RC8000</u>
a	A _ _ _ _ _	a
a27xx.obj	A 2 7 X X _ _ _ O B J	a27XX000obj
description	D E S C R I P T I O N	description
pip.7	P I P _ _ _ _ 7 _ _	pip000007
pip0.7	P I P 0 _ _ _ _ 7 _ _	pip000007

5.4 Catalog Entry Handling

5.4

The ACP process may create, change and delete files on RC8000. First of all: function send (reading the file) never changes the file, nor the catalog entry.

When receiving a file, the following actions are taken:

- Find a file with this name and having user base.
- If found (and not in use), use this file.
- If not found, create a new file with user base, of size = slice length (cf. section 2.2), on the disc stated in system trimming, and permanent the file.
- Receive the data, store it in file.
- If bytes transferred = 0 and the file was just created, it is removed again (note that the final status is irrelevant here).
- If the file is a text file, an EM (ISO 25) is inserted after last character.
- Unused part of last segment is filled with EM.
- If anything is written on the disc file (which may be avoided by interrupting), then
 - . bytes transferred is inserted in tail (10)
 - . shortclock is inserted in tail (6)
 - . the file is cut to the segments actually written (which is at least one).

In contrast to a binary transfer (where all bytes are transferred unchanged), text files are subject to character conversion, for two reasons:

- some characters are used by the transfer protocol, and should not occur in data.
- to facilitate the different "NEW LINE" conventions in CP/M and RC8000.

In CP/M a NEW LINE is CR (ISO 13) - LF (ISO 10).

In RC8000 a NEW LINE is NL (ISO 10).

When sending text files:

char values > 127: illegal (i.e. not text)

char values 0, 13, 127: blind (skipped)

char values 3, 16, 17, 19, 27: illegal (reserved)

char value 10 is converted to: 13 - 10

char value 25 ends the text.

All other values are sent without modification.

Illegal characters stops the transmission.

When receiving text files:

char values > 127: are impossible

char values 0, 13: blind (skipped)

char values 3, 16, 17, 19, 27: illegal (reserved)

char value 26: signals parity error (generated by the AMX).

char value 25: ends the text.

All other values are stored in the file without modification.

Illegal characters stop the transmission.

Parity error characters are counted.



A. REFERENCES

A.

- [1] RCSL No 31-D715:
RC8000 Asynkron Kommunikations Pakke (ACP8000)
Generel beskrivelse

- [2] RCSL No 31-D713:
RC8000 Asynkron Kommunikations Pakke (ACP8000)
Brugervejledning

- [3] RCSL No 31-D701:
RC8000 Shipping
General Information.

B. USE OF TEXTS IN ACPTRIM

B.

Long array txt:

Contains normal texts used in the terminal dialog. Note that all texts are written in mode 2. In the scheme below,

- SUP means that the text is suppressed (not written) if the wanted parameter is already stated.
- SKIP means that the text and a trailing value (and maybe another text) will not be written if the text is empty.
- FUNC tells in which functions the text is used, cf. section 5.1.

INDEX	WRITE	FUNC	USE
1			Prelude for user name and number at login.
2			Prelude for next command.
3	SUP	0,1,2,3	Prelude for RC8000 file name.
4	SUP	0,1,2,3	Prelude for terminal file name.
5	SUP	4	Prelude for drive.
6		4	used for checking drive: if specified drive equals txt(6), RC8000 is meant. Max 5 characters.
7	SUP	4	Prelude for file name mask.
8	SUP	5	Prelude for RC8000 file, that should be cleared.
9		6	Reply to command.
10		7	Reply to command.
11	SUP	8	Prelude for RC8000 file name, to print.
12	SUP	8	Prelude for printer name.
13		8	Used for default printer name. Max 5 chars.
14	SKIP	8	Reply to successful print request. Print ident is written after the text.

INDEX	WRITE	FUNC	USE
15	SKIP	8	Reply to unsuccessful print request. Result delivered (by procedure transfer) is written after the text.
16			Not used.
17			Reply to unknown command.
18	SKIP	6	Written if close is not accepted. (No of terminals logged in -1) is written after the text.
19	SKIP	6	Written after txt(18).
20		9	Terminal active in some function.
21		9	Terminal waiting for command.
22		0,2,4	Written before the terminal file catalog is received.
23		1,3,4	Written before the RC8000 main catalog is scanned.
24		0,2	First text, before sending a file. Followed by RC8000 file name and txt(25).
25		0,2	Second text, before sending a file. Followed by terminal file name and txt (26).
26		0,2	Third text, before sending af file.
27		1,3	First text, before receiving a file. Followed by terminal file name, and txt (28).
28		1,3	Second text, before receiving a file. Followed by RC8000 file name, and txt (29).
29		1,3	Third text, before receiving a file.
30		0,1,2,3	Fourth text, before any file transfer. Written after all checking of RC8000 files.
31		9	Terminal during function 1 or 3.
32		9	Terminal during function 0 or 2.
33		12	Reply (help).
34		9	Terminal during attention interrupt.

Long array err:

Contains texts written in error situations.

INDEX	FUNC	USE
1		Written at login, when user name and number is not found in user catalog.
2		Written when a command is recognized, function selected, but access mode does not permit use.
3	5,8	RC8000 file stated does not exist.
4	5	RC8000 file stated outside user base.
5	5	RC8000 file stated is in use.
6	5	RC8000 file stated could not be removed (after all).
7		Written at login attempts, if busy (maxterm terminals logged in).
8		Written at login, if user base cannot be reached by the ACP process.
9	0,1,2,3	User catalog contains file name, and RC8000 file name stated, is not this one. File name is written after the text.
10	0,1,2,3,4	Reading RC8000 main catalog is not possible p.t. as the common buffer is used by another terminal.
11		Abnormal internal alarm detected. Followed by 3 numbers: - 30 or trap-value - alarm cause extract 24: cause. Fx: - 15 killed (after operator break) - 11 give up (zone alarm) - 9 break - 2 index - 1 stack - alarm cause shift (-24): param.
12		At login attempt: get-specs-mess not answered ok. Probably because of wrong basis system.

C. START UP ALARMS FROM BACP

C.

All alarms described here use "provoked alarm" procedure in ALGOL, and they are recognized by the leading text and one parameter.

TEXT	EXPLANATION
messbufs -n	Too few message buffers. Add n and restart.
areaprocs -n	Too few area processes. Add n and restart.
bad cat n	Usercat not ok. n is $r*10+S$ r = result of reserveprocess S = 1: file not found, or not a user catalog 2: illegal records found.
bad wrk n	wrkfile could not be created. n = result of create entry.
newactiv o	Startup troubles, activities. Enough primary storage?
central n	Internal error in the activity monitor. n = result value of activate/new-activity.

D. PROCEDURE WRITESTATUS

D.

The procedure is called

- after termination of any file transfer
- after not-ok termination of a directory transfer.

Parameters:

zone z. Opened for the terminal, in mode 2.

integer res. Overall result of transfer

res extract 10: result value, see below.

res shift(-10): = 0 if all status blocks from terminal are
"STOK".

= 1 if any status block from terminal is <>
"STOK".

Status (3) and on contains the text received.

long array status.

status(1) : no of bytes in the file, i.e.: no of non-blind
bytes sent to the terminal or received and stored
on the disc - before any error occurred.

Status(2) : no of parity characters (ISO26) received from the
terminal.

Status(3) and on : last not-ok status block received, or
"STOK".

Below is a list of the possible values of res extract 10:

- 1 ok
- 2 terminal char illegal (fx control char in data).
- 3 discfile char illegal (>127 when text)
- 4 terminal char=26=parity received (status(2)=occurrences)
- 5 discfile char illegal (control char in text)
- 6 not used
- 7 checksum error detected (maybe set if otherwise ok)
- 8 terminal: get specs not ok.
- 9 not used
- 10 not used
- 11 createentrylockproc <>0, or checkbase<>0: file not found.
- 12 createentry=4 or permanententry<>0: claims exeeded

13 createentry=6: name illegal
14 createentry other results<>0
15 in use already, exclusive access wanted (i.e. write)
16 in use already, used exclusive (during write).
17 tail(1)<0: not datafile.
18 reserveprocess<>0 (at receive): reserved by another.
19 not used
20 not used
21 terminal status timer
22 terminal status attention
or trap: atmess arrived.
23 terminal giveup (hard error bit)
24 not used
25 file status=reject. reserved by another process (at send).
26 file status=end document, and func=write, and
changeentry<>0: no ressources
27 file status=does not exist (disappeared?)
28 other statuses/giveup
29 not used
30 program error (alarm not provoked by trap)



RETURN LETTER

Title:

RCSL No.: 31-D712

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

..... Fold here

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

Affix
postage
here

 **REGNECENTRALEN**
af 1979

Information Department
Lautrupbjerg 1
DK-2750 Ballerup
Denmark