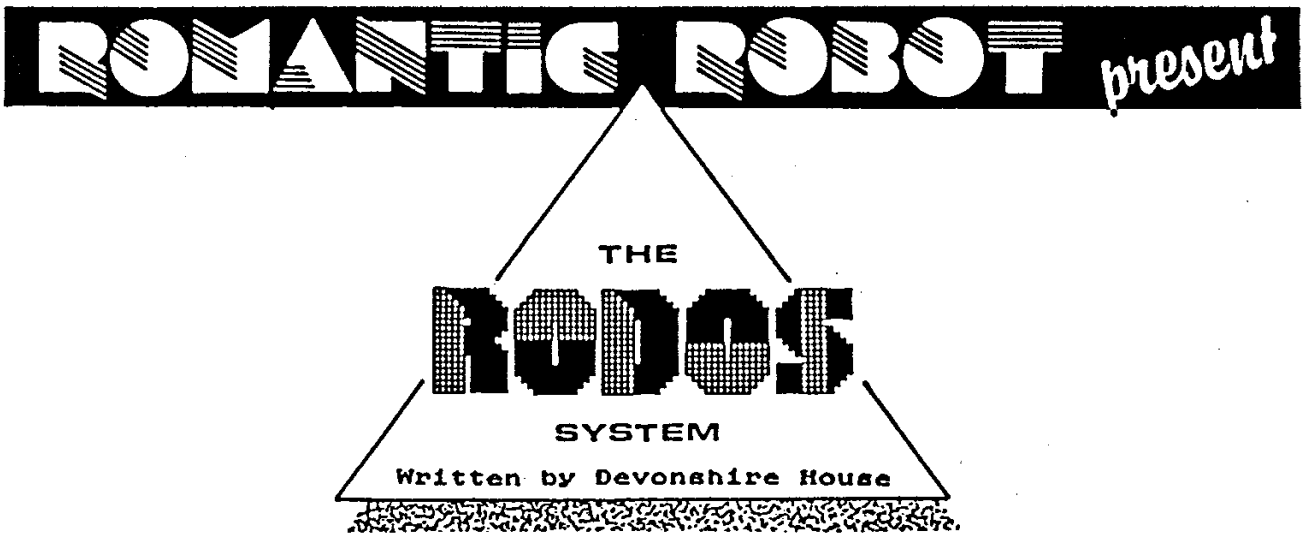

1. Introduction

RODOS and RECS are two powerful sets of utilities combined on one ROM
RODOS adds a whole new dimension to disk usage on the Amstrad, with

- MORE disk space available for the user
- UP TO 3 TIMES FASTER reading and writing to disk
- MORE EFFICIENT use of disk space, less waste
- SILICON disk facilities on 6128 or 464/664 with compatible RAM
- SUB-DIRECTORIES for organising your files logically
- REDEFINABLE DRIVES - drive letters A — H
- UNIX/MSDOS style command names
- EXTRA DISC DRIVES can be driven - 5 1/4 or 3 1/2 or 3 inch, 40/80 track, single/double sided
- RANDOM ACCESS files
- SPOOLER
- BUILT-IN DISK FORMATTER (no need for CPM)
- HEX DUMP, CHARACTER DUMP AND INFO ON FILES

RECS adds a host of new utility commands for you to use

- PRINTER BUFFER (up to 64k)
- CUSTOMISE bar commands
- BATCH FILES of bar commands
- COMMAND LINE INTERPRETER
- CHANGE PRIORITIES on ROMS
- DISABLE OTHER ROMS
- READ and WRITE sectors to other disks (eg. MSDOS & PCDOS format
- COMPATIBLE WITH EXTRA MEMORY (such as the Dk'tronics 256K RAM)

1.1 ROM Installation

This manual covers RODOS versions 2.15 and above

Please refer to the instructions for your ROM board for installation.
RODOS may be installed in any socket with higher priority than CPM (if present), i.e. 1,2,3,4,5 or 6; it must be a 16K socket.

Note :
1) Holding down the D key while the ROMS are initialised will stop RODOS from initialising - if for any reason it needs to be disabled. (Not 464's)

2) Holding down the R key while the ROM is initialised will reset all internal settings which are normally preserved through a reset.

3) Holding down the SPACE bar will stop auto-boot.
(Typing anything will also stop auto-boot on V2.15 and above)

1.2 Using RODOS and RECS

RODOS and RECS bar commands:

A	ACCESS	ALIAS	ASKRAM	B	BGET
BPUT	C	CAT	CD	CLI	CLS
COPY	DIR	DISC	DISC.IN	DISK.OUT	DISK
DISK.IN	DISK.OUT	DO	DRIVE	DUMP	EB
ERA	ERADIR	EXEC	FORMAT	FS	HELP
INFO	LIST	LOAD	LINK	LS	MD
MKDIR	MOTOR.OFF	MOTOR.ON	OPT	PEEK	POKE
POINT	PRINT	PRBUFF	RANDOM	READSECT	REN
RM	RMDIR	RODOS.OFF	ROMS	SAVE	SPOOL
TDUMP	TITLE	USER	WRITESECT	ZAP	

All RODOS and RECS commands are accessed by placing a bar '|' in front of the command,
eg. |DIR

Numbers 0-9 can be placed between the bar and the command (no spaces). This number is the ROM number (found using the |HELP command) to which you want the command to go. This allows you to send commands to ROMs which would normally be intercepted by ROMs with higher priorities.

Example: You have a Utility ROM in socket 6 and RODOS in socket 5. The Utility ROM has a command |PRINT which you wish to use, but RODOS has also got a |PRINT command and has a higher priority (lower socket No.) and therefore the RODOS |PRINT will always be chosen. If the command |6PRINT is given, then the Utility ROMs |PRINT will be selected (it is ROM number 6). Further modifications of ROM priorities are possible with RECS by using alias.

Example : |ALIAS,"HELP","5HELP"

This sets |HELP to do |HELP on ROM 5, regardless of any other ROM priorities.

See Also: |ALIAS

2. RODOS

RODOS is fully compatible with Amsdos. Normal BASIC command words such as LOAD, SAVE, OPENIN etc. use RODOS. RODOS will work with Amsdos formatted disks as per normal use (RODOS is transparent to the Amsdos disk user), although many of the features of RODOS, such as directories will not be available while using Amsdos formatted disks. Disks formatted under RODOS open up a whole range of features normally found on only larger, more expensive machines. Plus the added bonus of faster access and more efficient use of disk space.

2.1 Directories

A directory is a group of files on a disk. Under Amsdos a disk may be split into 'user' sections, which are separate directories. These are somewhat awkward to use and impossible to structure in any way.

RODOS gives a fully structured hierarchical directory with directories and their sub-directories. Each directory has a name, in much the same way as a file. The main directory is called the root directory and has a special name '/'. This corresponds to the one and only directory you have when you do a |DIR on an Amsdos disk. In RODOS, you may create further directories (sub-directories) in the main directories.

Consider a disk of BASIC games. There are many types of game on the disk and they are mixed together in storage order when doing a |DIR.

```
SPELL      RW        CYCLE      RW        OXO        RW        MAZE       RW
KILLER     RW        SUNDOG     RW        WHODUNIT   RW        COLOSSAL   RW
FIGHTER    RW        PAWN       RW
```

(The RW indicates the file is a read/write file.)

It is unclear which games are adventures, strategy or arcade style.
With RODOS command MKDIR you can create three sub-directories: arcade, adventure, and strategy and place the games in the appropriate ones.

Doing a |DIR (under RODOS) you get

```
adventure  Dir              strategy Dir              arcade   Dir
```

You can now use the |DIR command in a new way, to look at the contents of these sub-directories.

```
|DIR,"/strategy/"
```

Gives us:

```
OXO    RW          MAZE   RW          SPELL  RW
```

You can also load files by prefixing their names with the directory path (a path leads through all sub-directories to the file).

```
LOAD "/strategy/oxo"
```

If you intend spending a lot of time looking at files in a directory we can 'move into' the directory, making it your current directory:

```
|CD,"/strategy"
```

(or |CD,"strategy" if default directory is already root).
Files can now be loaded or saved in this directory without needing a path prefix.
To move back to the root directory you can use one of two methods:

```
|CD,"/" will set the current directory to root.
|CD,".." will set the current directory to the one above this one
 (you are assumed to go 'down' when entering a directory).
```

2.2 RODOS commands.

The command descriptions are split up as follows:

NAME - name of command
SYNOPSIS - command usage with parameters
DESCRIPTION - what it does
See Also - other related commands

The following terminology is used in the command descriptions:

FILENAME - These are names up to 16 characters long.
If a full stop is used the name is split into
12 characters, '.' and a 4 character extension.
PATH - A prefix of sub-directories, showing the path to
the required file or directory.
DIRECTORY - A directory name has the same format as a file name.
WILDCARD - A wildcard filename will match the FIRST matching entry
only (unless otherwise specified). All filenames may
include wildcards unless the file is to be created.

When parameters are given in quotes, CPC 464 owners must place the quoted string in a string variable and pass the string variable instead (unless using the CLI).

Eg. Instead of !ERA,"filename" you must type

```
a$="filename"
!ERA,@a$
```

****

A B C DRIVE

NAME DRIVE - select current default drive
A - select drive A
B - select drive B
C - select drive C

SYNOPSIS

```
!DRIVE,"drive_letter"
!DRIVE,"drive_letter",drive_number
!A
!B
!C
```

DESCRIPTION

With a drive letter (in quotes - both upper or lower case) sets the given drive as the default drive.
Eg. !DRIVE,"B" sets the default drive to drive B.

With an additional drive number, this command sets a 'logical' drive letter to a 'physical' drive. i.e. the drive given by the drive number will be accessed by the drive letter given.
Eg. !DRIVE,"B",0 sets "B" to refer to the physical drive 0 (the built-in drive of a 6128 or 664 - see Appendix A for a list of drive numbers, see Appendix B for drive letter to number initial values).

Commands !A !B !C are shorthand versions of !DRIVE,"drive_letter": they set a current default drive (!A sets default drive to A etc.)
!A, !B, !C ignore any parameters given on V2.15 and above)

See Also: !DIR

****

BGET BPUT

NAME BGET - Get a byte from an open file: to an existing variable (integer type.)
BPUT - Put a byte to the open file.

SYNOPSIS !BGET,@integer_variable
!BPUT,character_number

DESCRIPTION

BGET One byte is fetched from the file and placed in the given integer variable. Eg. !BGET,@a% The variable used must already exist and be an integer. If a number less than zero is returned there will be an error message (such as end of file).

BPUT Writes the given character to the open file.

Eg. !BPUT,asc("B") will put a "B" into the file.

See Also: !POINT, !RANDOM

****

CD

NAME CD - Change Directory.

SYNOPSIS !CD,directory_name

DESCRIPTION

Sets the current working directory to the given directory. The name may contain a list of sub-directories separated by a slash (known as a path). All names must be directories, not files.

Eg. !CD,"/games/space" sets the current directory to sub-directory space which is in games which is in the root directory '/'.

Directory names may also be given in 'relative terms' i.e. relation to the current directory.
Eg. !CD,"letters" sets the current directory to letters which is then the current directory. It is therefore possible to step down easily through the branches of the sub-directories.

A special directory name '..' means the 'parent' directory
(the directory above this one).
Eg. !CD,".." will set the current directory to the one above.
(you cannot go above / (root)).
!CD,"../stuff" translates to: 'move up 1 to parent, then go into stuff'

All references to files will assume the file is in the current directory unless a directory path is specified in the filename.

If no directory_name is given, RODOS will change to a default directory which is the directory and drive last changed to with
!DRIVE or !A, !B or !C

Example:
!CD
!CD,"B:/tmp"

****

CAT

NAME CAT - Catalogue disc (or other file system)

SYNOPSIS
```
!CAT
!CAT,"file_system"
```

DESCRIPTION

Displays the files on a disk as the immediate BASIC command CAT.

!CAT with a file system name will perform CAT on that system
Eg. !CAT,"TAPE" will do a CAT of a tape.

See Also: !DIR

****

COPY

NAME COPY - Copy a file

SYNOPSIS
```
!COPY,"newfile","oldfile"
```

DESCRIPTION

This command reads one file and writes it to another.
Eg. !COPY, "-TAPE-DOCUMENT.BAK","B:PAGES" will copy the file "PAGES" from drive B to tape under the name of "DOCUMENT.BAK"
If a wildcard is used in the name, all the names matching the filename with wildcards are copied, and the file name found is added to the end of the new name. This allows across drive copying.
Eg. !COPY,"B:","*.DOC" or !COPY,"-TAPE-","*"

Note: when copying lots of files, it is recommended to use !OPT,1,255 first, as this will display a list of the files as it copies them. Also, when using !COPY, RODOS uses memory from &8000 to &9000 as buffers, any existing data in these areas will be lost.

****

DIR

NAME DIR - List a directories contents.

SYNOPSIS
```
!DIR
!DIR,"filename"
!DIR,"directory/[directory/][filename]"
```

DESCRIPTION

On its own, !DIR will list the files (and sub-directories) of the current directory.

With the optional filename, it lists all the files matching the specified name. The filename may include a directory path (see CD).

Eg. !DIR,"letters/jimmy.doc" lists the file 'jimmy.doc' in the sub-directory 'letters', if it exists.
!DIR,"letters/jimmy.*" lists all files starting with "jimmy"in the-directory called letters.

With a directory name, it lists the files (and sub-directories) in that directory. The last directory name must end with character '/' to show it is a directory and not a file name.

Eg. !DIR,"/games/adventures/" lists files in the sub-directory 'adventures', which is found in the sub-directory 'games'.

See Also: !CD, !CAT, !DRIVE

****

DISC or DISK

NAME DISK - Select disk filing system for input and output (as opposed to tape etc.)
DISK.IN - Select disk filing system for input (output unchanged - eg. tape for input, disk for output.)
DISK.OUT - Select disk filing system for input and output.

DISC - As for DISK
DISC.IN - As for DISK.IN
DISC.OUT - As for DISK.OUT

SYNOPSIS !DISK
!DISK.IN
!DISK.OUT

DESCRIPTION

!DISK selects disk as the current filing system for input & output.
!DISK.IN selects disk as the filing system for input. The former filing system remains in use for output.

Eg. !TAPE
!DISK.IN

sets filing system to tape, then resets input for disk allowing input from disk and output to tape.
!DISK.OUT selects disk as the filing system for output. The former filing system remains in use for input.

Eg. !TAPE
!DISK.OUT

sets the filing system to tape, then resets output for disk, allowing input from tape and output to disk.

****

DUMP

NAME DUMP - Hex dump a file.

SYNOPSIS !DUMP,"filename"
!DUMP
!DUMP, “filename" ,number

DESCRIPTION

Dumps a hexadecimal listing of the given file to the screen. When a number is specified, the dump address starts at this number.
For printer output use the !PRINT command as well (see RECS).

****

EB

NAME EB - Erase Backups.

SYNOPSIS !EB

DESCRIPTION

All files in the current directory with the extension .BAK are deleted.
Equivalent to: !ERA, "* .BAK"

See Also: !ERA

****

ERA, RM

NAME ERA - Erase a file (or files if wildcard used in name).
RM - As for ERA

SYNOPSIS
```
!ERA,"filename"     !RM,"filename"
!ERA                !RM
```

DESCRIPTION

The given file is deleted (will not delete directories. though).
The filename may include directory information.
Eg. !ERA,"games/game.duf" deletes the file 'game.duf' in the directory games.
You must have write permission on the file to do this (see Access). If no filename is given, one is asked for.

See Also: !ERADIR, !DIR, !CD

****

ERADIR, RMDIR

NAME ERADIR - Erase Directory
RMDIR - As for ERADIR

```
!ERADIR,"directory"          !RMDIR,"directory"
!ERADIR                      !RMDIR
```

DESCRIPTION

The given directory is deleted provided it is empty.

See Also: !ERA, !CD, !DIR, !MKDIR

****

FORMAT

NAME FORMAT - Format a disk

SYNOPSIS
```
!FORMAT,type
!FORMAT,type,drive_number
!FORMAT,type,drive_number,tracks
!FORMAT,type,drive_number,tracks,sides
```

DESCRIPTION

Formats a disk in one of the standard formats given by type:

0 - IBM (Amstrad version) 2 - RODOS format

1 - CPM (Amstrad version) 3 - DATA (Amstrad version)

The optional drive number is the physical drive to format (see Appendix for drive numbers). Otherwise the default drive is assumed.
'tracks' refers to the number of tracks on the disk (1-80).
'sides' refers to the number of sides on the disk (1-2).

When formatting the silicon disc, the default number of tracks takes ALL of the unused memory banks. If less memory is required then the number of tracks can be calculated: each track consists of 10 sectors of 512 bytes each, i.e. each track requires 5K.

NOTE: Printer buffer must be set before formatting the silicon disc if to be used in the extra Ram, so that the silicon disc can take all the memory UP-TO the printer buffer. without any memory clash.

FORMAT (contd.)

TECHNICAL INFO

When the type is 0,1 or 3 (i.e. normal formats) only 40 tracks should be used (numbered 0-39), The AMSDOS system assumes this number, and can only access side 0. RODOS allows for any number of tracks 1-255, and can use both sides. All four formats use sectors of 512 bytes, but RODOS has 10/20 sectors per track [numbered #81-#8A (single sided) or #81-#94 (double sided)], where CPM and DATA formats have 9 [i.e. numbered #41-#49(CPM format) or #C1-#C9 (Data format), and IBM has 8 (numbered #01-#08)].

(See appendix C for information layout on the disc.)

The sectors are numbered from #81 to #8a on side one of a disc, and #8B to #94 on side two of the disc. Tracks are numbered 0-39 on a 40 track disc, and 0-79 on an 80 track disc.

Please note that with RODOS, a disc can be formatted to any number of tracks (1 to 255) and NO checks are made that the disc actually has this number. Note also that most disc drives actually allow up to 2 or 3 more tracks than they specify, ie 42 tracks on a 40 track disc, or 82 tracks on an 80 track disc.

!FORMAT,2,0 formats 40 tracks (0-39), on one side (sectors #81 to #8A) [200k]

!FORMAT,2,1,82,2 formats 82 tracks (0-79), on two sides (sectors from #81 to #8a on side one, and #8F to #94 on side two) [820k]

****

INFO

NAME INFO - File information.

SYNOPSIS

```
!INFO,"filename"
!INFO
!INFO,"filename" ,@t%,@S%, &L%, x%
            Where:
                  t% = type
                  S% = start
                  L% = length
                  x% = execute address
```

DESCRIPTION

Gives information on files in the form:

name type load_address + size x execute_address

Where:

name = filename

type = one of the following special characters:

$ - unprotected BASIC file.

% - protected BASIC file.

* - ASCII file.

& - Binary file

load_address = load address in hex

+ size = size (in bytes)

x execute_address = execute address (in hex)

If no filename is specified, all files will be used. If followed by 4 variable addresses, it puts info on the file into these variables.

See Also: !DIR, !CAT, !DUMP, !LIST

****

LINK

NAME LINK - Duplicate a directory entry

SYNOPSIS !LINK,"path/filename"

DESCRIPTION

Given a filename (which does not exist in the current directory), this will put a duplicate entry in the current directory. NOTE this is not another copy of the file BUT THE SAME FILE in the same place on the disc, and thus not taking up any more disc space. The link CAN be to a different drive, and if so, the drive number 0-255 is stored in the entry: to use the file, the original disc must be in the same place.
Erasing a linked entry will not erase the original file.
Erasing the original file will invalidate all its links.

WARNING - the new entry holds the position on the disc of the file, so if it is deleted (from its original entry), or saved over, then the positional information will be wrong, and the linked entry will not be valid. It is therefore recommended to use links with care,
such as from the silicon disc to your files on your drives.

A good use for this command is to set up a silicon disc as a main working drive, and then link in such things as a utilities directory from a normal drive, thus giving you your utilities inside your silicon disc (like putting one disc inside another).

Example of use

!CD,"C:" (switch to drive C)
!LINK,"A:/utils/assembler"

this puts a file entry called "assembler™ on your C drive, which when loaded will refer to your A drive file.

****

LIST

NAME LIST - List a text file.

SYNOPSIS !LIST,"filename"
!LIST

DESCRIPTION

This lists the given file name to the screen. If no name is given, one is asked for. Printer output can be achieved via !PRINT.

See Also: !INFO, !DUMP

****

LOAD

NAME LOAD - Load a file

SYNOPSIS
```
!LOAD."filename"
!LOAD, "filename",bank
!LOAD,"filename",start,length
```

DESCRIPTION

Given just a file name the file loads at its normal load address.
With a bank, the file name is loaded, starting at the base of the given bank.
With a start and length, the file is loaded, starting at the memory address given by 'start'.

See Also: !SAVE

****

LS

NAME LS - List files

SYNOPSIS
```
!LS
```

DESCRIPTION

UNIX style version of DIR. No file or directory may be specified.

See Also: !DIR, !CAT

****

MKDIR MD

NAME MKDIR - Make directory.
MD - as MKDIR

SYNOPSIS
```
!MKDIR,"directory"
!MKDIR
!MD, "directory"
!MD
```

DESCRIPTION

Creates a new sub-directory with the given name.

Eg. !MKDIR,"project" creates the sub-directory 'project' in the current directory.

The name may include the path name.
Eg. !MKDIR,"project/old" will create a sub-directory 'old' in the sub-directory 'project' which is in the current directory.

!MD is a short version of !MKDIR.

NOTE: Both MKDIR and MD will only work on disks using the RODOS format. if no name is given, one will be asked for.

See Also: !CD, !ERADIR

****

POINT

NAME POINT - move file pointer

SYNOPSIS
```
!POINT,position_number
!POINT ,number ,@integer_variable
```

DESCRIPTION

Moves the current read position in a file (and the write position if random access).
If "!POINT,1,@a%" is used, a% will return the current place in the file - can be used after !POINT,65535 to get the length of a file.

```
Eg. 10 OPENIN "datafile"
20 !POINT,20
30 a%=0: !BGET,@a%
40 PRINT "The 20th byte of the file is ";a%
```

If the position number is past the end of the file, the read/write position will move to the file end. This can be used to move to the end of a file - and by using !RANDOM add extra things to it.

```
Eg. 10 OPENIN "datafile"
20 !POINT,65535: REM **past the end of the file**
30 !RANDOM
40 PRINT#9,"add this data"
50 CLOSEIN
```

Note: This will only work with RODOS format discs.

See Also: !RANDOM, !BGET, !BPUT

****

RANDOM

NAME RANDOM - declare a random access file

SYNOPSIS !RANDOM

DESCRIPTION

Opens a currently open INPUT file for writing. Any output to the file (ie print #9) will overwrite part of the existing file (or add to the end). CLOSEOUT will return the file to an input file, and write any changes out to disc. CLOSEIN will write any changes to
the disc and close the buffer.

```
Eg.    10 OPENIN "datafile"
       20 !RANDOM
       30 INPUT "record number ";rec%
       40 !POINT,rec%*7
       50 INPUT#9,d$:PRINT "old data ":d$
       60 INPUT "new data":n$
       70 IF LEN(n$)>5 THEN 60
       80 IF LEN(n$)<5 THEN n$=n$+" ":GOTO 80
       90 !POINT,rec%*7
       100 PRINT#9,n$
       110 GOTO 30
```

Will allow changes in the middle of a file with entries made up of up to 5 characters in length (total length of 7 incl. CR and LF).

See Also: !POINT, !BGET, !BPUT

****

REN

NAME REN - Rename a file (or group of files).

SYNOPSIS |REN,new_filename,old_filename

DESCRIPTION

Renames a file. Directory paths can be used on the old_filename
Wildcard filenames will match ALL occurrences in the current directory.
Eg. |REN, "* BAK","*.BIN" will rename ALL occurrences of files ending with .BIN and change them to .BAK.
Note : Wildcards CANNOT be used with REN on AMSDOS discs.

See Also: |DIR, |ERA

****

TITLE

NAME TITLE

SYNOPSIS |TITLE,"name"
|TITLE

DESCRIPTION

This sets a disk title/label on the disk (RODOS format only).
The name can be up to 16 characters and include control characters.
Eg. |TITLE,"Games Disk" or |TITLE,chr$(12)+"Fancy Dir"
If no name is given, one is asked for. If the name is "" then the title is set to all spaces.

See Also: |FORMAT

****

USER

NAME USER - Sets user number.

SYNOPSIS |USER,number
|USER

DESCRIPTION

As CPM USER command (see Amsdos Manual - CPM)

Note: RODOS supports user numbers 0-65535, Amsdos only 0-15. RODOS displays all files in a directory, Amsdos only displays those owned by the correct user.
User numbers entered by |USER have &ff00 added to them, so only numbers 0 to 255 can be entered.
(other user ID's are reserved for further expansion & logins)
The user number entered is logically added with &0F and passed to the CPM |USER.

|USER without a number is as |USER,0

****

3. RECS

Whether you use RODOS or not (and you should!) you have at your disposal a set of powerful commands in the ROM Extended Command System (RECS for short). These varied commands extend the operating system to provide flexible control over the printer, screen, RAM and ROMs. These commands include printer buffer, ROM command aliasing, ROM priority shuffling, individual ocommand priority changes, text screen dumps, redirection of screen output to printer or file or both!
No longer will 464 owners have to struggle with assigning string variables for ROM commands. SHIFT & FUNCTION-ENTER puts you straight into the superb Command Line Interpreter. For example on the 464, renaming a file can be a tiresome prospect:

```
n$="newfile"
o$="oldfile"
|REN,@n$,@o$
```

But in the CLI....

```
SHIFT & FUNCTION-ENTER (1 press to enter the CLI)
REN newfile oldfile
```

Ever had this problem:

```
10 input a$
20 IF a$="CAT" THEN CAT
30 IF a$="TAPE" THEN |TAPE
40 IF a$="DISC" THEN |DISC
50 IF a$="A" THEN |DRIVE,"A"
60 etc.
9999 GOTO 10
```

Why not use the |DO command of RECS:

```
10 input a$
20 |DO,a$
```

(Plus the RECS version can take parameters!)

With RECS you can make more of your machine!

3.1 RECS Commands

The same conventions are used as for RODOS:

NAME - name of command
SYNOPSIS - command usage with parameters
DESCRIPTION - what it does
See Also - other related commands

FILENAME - These are names up to 16 characters long. If a full stop is used the name is split into 12 characters, '.' and a 4 character extension.

PATH - A prefix of sub-directories, showing the path to the required file or directory.

DIRECTORY - A directory name has the same format as a file name.
WILDCARD - A wildcard filename will match the FIRST matching entry only (unless otherwise specified). All filenames may include wildcards unless the entry is being created on the disc.

Also: Holding SHIFT and pressing CAPS LOCK will do a SHIFT LOCK.
Holding SHIFT and CONTROL will pause any screen output.

If a disc with a program called DISC is present in drive "A" or "B". RECS will automatically run it at switch on or reset (unless SPACE is pressed during the reset, or any key is pressed in the first few seconds).

****

ACCESS

NAME ACCESS - Set file protection.

SYNOPSIS !ACCESS,"filename",mode_number

DESCRIPTION (RODOS discs)

Changes read and write modes on the given files to mode_number where mode number is the sum of:

1 - Read Permission	8 - Read Permission (other users)
2 - Write Permission	16 - Write Permission (other users)
4 - Execute only	32 - Execute only (other users)

Note: With 4 and 32, Read Permission should also be given (1 or 8).
Read/Write permissions are shown as "R" or "W" during a DIR or CAT.
Execute protect is shown as "X" allowing a program to be loaded or run, but not listed or dumped. If set on a basic program saved with " P" on the end, it allows it to be run only. Also note that you must be the owning user in order to change access on a file.

ACCESS,"myfile.bas",27 gives read & write permission to all users
(27 = 16 + 8 + 2 + 1).
ACCESS,"myfile*",45 gives all files starting by "myfile" Execute only.
(45 = 32 + 8 + 4 + 1).

DESCRIPTION (AMSDOS discs)

Changes Read/Only and SYStem status of files. The mode number is:

0 - Normal file
1 - SYS (System file. not shown in the directory)
2 - R/O (Read Only, file cannot be erased)
3 - SYS and R/O

See Also: !DIR

****

ALIAS

NAME ALIAS - Define new command name as old
(with optional parameters)

SYNOPSIS !ALIAS,new_command_name,old_command_name
!ALIAS

DESCRIPTION

The new_command_name will be translated to the old_command_name before it is processed. This allows "customisation" of ANY bar commands to be anything you want
Eg. !ALIAS,"H","HELP 7"
!ALIAS,"DEL","ERA TEMP"
lets you use !H instead of !HELP.7 and !DEL instead of !ERA,"TEMP"

If used without parameters, the command will display all the new command names together with their actual names and parameters.

The new_command_name must not be identical to the old_command_name; you must not ALIAS a command to be itself. If you wish to do this,use the ROM_number prefix:
i.e. !ALIAS,"DIR","7DIR *.BAS"
Parameters may be referred to as @1 to @9

i.e. !ALIAS,"DIRECTORY","DIR @1".
Parameters are NOT checked by RODOS

****

ASKRAM

NAME ASKRAM - Amount of available RAM.

SYNOPSIS !ASKRAM.@integer_variable
!ASKRAM,number,@integer_variable

DESCRIPTION
Sets the integer variable to the number of available Kbytes.
Eg. 10 a%=0
20 !ASKRAM,@a%
30 PRINT "there is ";a%;"k available"

Alternatively, If ASKRAM is followed by number 1, the variable is set to the number of 16K banks.
Note: This Ram count is of extra Ram, and does not include the normal 64K. (i.e. it would normally give 0 on a 464/664 and 4 on a 6128)

See Also: !PRBUFF, !FORMAT
****

CLS

NAME CLS - Clear Screen

SYNOPSIS !CLS

DESCRIPTION
Clears screen to mode 2 with white text (ink No.13) on black paper.

****

CLI, DO

NAME CLI - Command Line Interpreter.

SYNOPSIS !CLI
!CLI,"command_string"
!DO - as !CLI

DESCRIPTION
Without a string parameter, a bar prompt will be given. Bar commands can be entered without being preceded by bar and will be executed.
Entering a blank line will exit the CLI. While in the CLI, parameters may have either a comma or a space as a separator.
Numeric values can be entered in hex preceded by a hash or an ampersand.
Strings (without spaces) may be entered without quotes (provided the first character is not a numeric. hash or ampersand).

Eg. 10 REM basic program
20 PRINT "Enter bar commands:"
30 !CLI : REM user terminates with a blank line
40 PRINT "Continuing"

A string parameter must hold a valid bar command. The bar command will be executed once (returns to Basic).
Eg. 10 a$="DIR"
20 INPUT b$
30 c$=a$ + b$
40 !DO,@c$

See Also: !HELP
****

EXEC

NAME EXEC - Execute a batch command file.

SYNOPSIS !EXEC,"filename"
!EXEC

DESCRIPTION

This loads and executes an ASCII file as a series of bar commands.
Any line in the file starting with a bar character are passed to CLI (see !CLI). Lines without a bar at the start are sent to the screen.
Here is a simple example of a startup command file:

```
!CLS
!PRBUFF
!ALIAS,H,HELP
Setting up Silicon Disc
!FORMAT,2,8
!DRIVE,"A",8
!A
Working in Silicon disc (drive A).
Set-up Done!
```

Note: If any of the Bar commands open a file for input, such as !List, the EXEC will stop. This is because there is only one file input channel, which would have just been closed.

It is useful to !ALIAS a bar command to !EXEC a particular file.
Eg.: !ALIAS,"S","EXEC startup-file"

Any parameters given on the alias could be picked out in the exec file as @1 to @9.

See Also: !CLI

****

FS

NAME FS - Filing System.

SYNOPSIS !FS

DESCRIPTION

Adds the option of using a filing system specified in front of a file name while still in a different filing system (eg. Tape).
Eg.
```
!TAPE
!FS
LOAD "-disc-game"
```
will load game from disc while remaining in the tape filing system.
Note: This is automatically done by !DISC or !DISK, but not !TAPE, etc.

See Also: !DISC, !DISK

****

HELP

SYNOPSIS !HELP
!HELP, number

DESCRIPTION

!HELP alone will list all ROMs in sockets 0-15 with their numbers.
When used with a ROM number, it will list all its bar commands.
Eg. !HELP,1 will list all bar commands on ROM 1.

See Also: !ROMS, !ZAP

****

MOTOR.ON **MOTOR.OFF**

NAME MOTOR.ON - Turn tape motor on.
MOTOR.OFF - Turn tape motor off.

SYNOPSIS !MOTOR.ON
!MOTOR.OFF

DESCRIPTION
Controls the tape motor, i.e. turns it ON and OFF.

****

OPT

NAME OPT - Sets options.

SYNOPSIS !OPT,option_number,value

DESCRIPTION
This powerful command allows you to set several RODOS defaults:

```
1,n    - loading messages (on/off)
2,n    - case sensitivity on file names (on/off)
3,n    - cassette loading messages {(on/off)
4,n    - overwrite file. n=0, 1 or 2
             0 - ask
             1 - erase (overwrite)
             2 - create backup
5.n    - disk read error retry count (default=16)
6,n    - disk error messages (on/off)
7,nn   - motor on time in 1/50 sec. (default=11 sec)
8.nn   - motor off time in 1/50 sec. (default=7 sec)
9,n    - drive tracking speed in ms. (default=12)
10.n   - head load delay in ms. (default=1)
11,n   - head unload delay in ms. (default=1)
12,nn  - extra external disk drives port number
13,n   - enables 40 track disk to be read on 80 track drive
         in double step (on/off)
14,n   - head settle time in ms (default 15)
```

Values:
n = single byte value (0-255)
nn = double byte value (0-65535)
on = 255
off= 0

****

PEEK **POKE**

NAME PEEK - Peeks a byte from memory.
POKE - Pokes a byte into memory.

SYNOPSIS !PEEK,bank,address,@integer_variable
!POKE bank,address,integer

DESCRIPTION
PEEK peeks the contents of the memory location given by 'address' and 'bank', and places it in the 'integer variable'.
POKE pokes the value of 'integer' into the memory location given by 'address' and 'bank'. The address in both commands is the address within the bank (0000 - 3FFF).
Useful only for the 6128 or for 464/664 with extra memory.

****

PRINT

NAME PRINT - Screen output -redirection.

SYNOPSIS !PRINT,n
!PRINT

DESCRIPTION

Sets normal screen output according to n, where n is the sum of the following:

+1 - screen output to current open file.
+2 - screen output to printer.
+4 - screen echo disabled.

These values can be added together in any combination.

Eg. !PRINT,2
CAT gives a hard copy of the files on the disk or tape.
!PRINT,6 = 2+4 = All screen output to printer.

!PRINT on its own is the equivalent of !PRINT,0 - i.e. it resets screen output to its normal state.

See Also: !PRBUFF, !TDUMP, !SPOOL

****

PRBUFF

NAME PRBUFF - Sets up a printer buffer.

SYNOPSIS !PRBUFF
!PRBUFF,bank
!PRBUFF,start,end
!PRBUFF,start,end,bank

DESCRIPTION

This command sets up a printer buffer which allows printing without holding up the computer. A listing can be sent to the printer, then while it is being printed you can work on something else.

!PRBUFF on its own creates a default printer buffer which is 16K (one 16K bank).

NOTE: If formatting a silicon disc, PRBUFF should be done first, otherwise the silicon disc will take all the RAM.

When only one number is specified, it is taken as the memory bank number to be used as the print buffer (the whole 16K bank is used).
When two parameters are specified, these are taken as the start and end addresses of the printer buffer in normal RAM (464/664 use).
When three parameters are specified, the addresses given by 'start' and 'end' are taken to be addresses within the given bank.

Note: Addresses greater than #3FFF (16383) overlap into the next bank.
I.e. !PRBUFF,0.65535,1 will set up a 64k printer buffer on the 6128.

Eg. 10 MEMORY 33000
20 !PRBUFF,33000,40000

See Also: !PRINT, !TDUMP

****

READSECT

NAME READSECT - Read a physical sector.

SYNOPSIS !READSECT,address,drive_number,track,sector

DESCRIPTION

READSECT allows the direct reading of a physical sector on a disk into memory. It enables you to read disks formatted under other operating systems such as MSDOS. Use cautiously!

```
     address = memory location of 512 byte buffer to hold the sector
drive_number = physical drive number (appendix A)
       track = physical track (0-160)
      sector = physical sector number (+256 for the other side)
```

See Also: !WRITESECT, !OPT

****

RODOS.OFF

NAME RODOS.OFF - Disable RODOS with an optional start up string

SYNOPSIS !RODOS.OFF
!RODOS.OFF,"start up string"
!RODOS.OFF,""

DESCRIPTION

This disables RODOS and restarts with a start up string. This command is designed to assist with games and applications that require more memory than is available with RODOS active.

NOTE: Only available in RODOS V2.22 and later

See Also: !ROMS, !ZAP

****

ROMS

NAME ROMS - ROM priority.

SYNOPSIS !ROMS,rom_number [,rom_number]...
!ROMS
!ROMS,rom_number [,rom_number]...,"start up string"

DESCRIPTION

This command "Turns On" the given ROMS. The normal ROM priorities are overidden by the ordering given in the list of ROM numbers. Any ROMS in sockets that are not in the given list are turned off.
!ROMS on its own disables all ROMS except CPM ROM (if in socket 7).

NOTE: !ROMS restarts the current language and any program is lost.
!ROMS with a start up string will cause the start up string to be typed after the reset.
E.g. !ROMS,2,3,"F"+chr$(13) will type "F" and press return after resetting the machine and initialising Roms 2 and 3.

464 USERS!: RECS allows 15 ROMS to be used (numbers 0 to 14). When the 464 is initialised ROMS 14-8 are initialised before RODOS.

See Also: !ZAP, !HELP

****

SAVE

NAME SAVE - Save a block of memory.

SYNOPSIS !SAVE,filename,start,length
!SAVE,filename,bank
!SAVE,filename,bank,start,length

DESCRIPTION

The first alternative saves a variable size block of memory to disk.

filename = name of file to save.
start = start address in memory (0 to 65535).
length = number of bytes to save (0 to 65535).

The second form specifies a bank number (16K memory block) to save and is useful on a 6128 or on an expanded 464 or 664.
Valid bank numbers are:

0 - first 64K bank (4000-7FFF)
1 - second 64K bank 1.
2 - second 64K bank 2.
3 - second 64K bank 3
4 - second 64K bank 4
5+ - further RAM expansions

****

SPOOL

NAME SPOOL

SYNOPSIS !SPOOL,"filename"
!SPOOL

DESCRIPTION

!SPOOL will close the file and reset the screen output to normal. With a filename, !SPOOL opens a file with that name (for output) and sets the screen output to printer (see !PRINT,2). Further screen output will be copied into the file.

Note: DIR and CAT close the output file and therefore SPOOL will Not work with them.
Use !LS instead (RODOS-only)

See Also: !PRINT

****

TDUMP

NAME TDUMP - Text dump of screen to printer.

SYNOPSIS !TDUMP

DESCRIPTION

Sends a straight text dump of the screen to the printer. Non characters on screen are treated as spaces.

See Also: !PRBUFF, !SPOOL

****

WRITESECT

NAME WRITESECT - Write a physical sector.

SYNOPSIS !WRITESECT,address,drive_number,track,sector

DESCRIPTION

This powerful command allows direct writing to physical sectors on a disk. As with Readsect, other format disks can be written to.

```
    Address  = memory location of a 512 byte buffer containing the data to be
               written.
drive_number  = physical drive number (appendix A)
      track   = physical track (0-160)
      sector  = physical sector number (+256 for the other side)
```

WARNING: This command can destroy file information and directories and as a result make normal use of the disk impossible. Use with care!

See Also: !READSECT, !OPT

****

ZAP

NAME ZAP - Disables ROMs

SYNOPSIS
```
!ZAP,rom_number [,rom number]
!ZAP
!ZAP,rom_number,...,"startup"
```

DESCRIPTION

This disables the ROMs in the given list. All other ROMs are enabled in normal priority order.
!ZAP on its own disables all ROMS.

NOTE: !ZAP restarts the current language.
!ZAP with "startup" will cause "startup" to be typed at reset.
Eg. !ZAP,3,"RUN"+CHR$(34)+"PROG"+CHR$(13) will disable Rom 3 and run a program called PROG.

See Also: !ROMS, !HELP

****

APPENDIX A:
PHYSICAL DRIVE NUMBERS

0	Normal drive A
1	Normal second drive
5	Normal drive B (side 2)
8	Silicon Disc (2nd 64k onwards)
9 - 199	Extra external drives
200 - 219	Reserved for silicon discs
220 - 254	Reserved for Hard Discs
255	No drive

****

APPENDIX B:
DEFAULT DRIVE SETTINGS

Disks are referred to by the system by their number, which is converted from their letter ("A" - "H"). So any drive letter can refer to any drive number, and therefore to any real drive.
Note: This does not apply to programs which use there own file loaders.

	Drive letter	— real drive	
	A	0	(internal of 664/6128)
	B	9	(normal 2nd drive or 1st exp drive)
	C	8	Silicon disc
	D	10	2nd expansion drive
	E	11	3rd expansion drive
	F	255	no drive
	G	255	no drive
	H	255	no drive
Note	H	220	Hard disc on version 2.15 and above

The above numbers are held in a table at address &BE00, which ends with a '99' byte at &BE09 to indicates its presence.

****

APPENDIX C:
DISC LAYOUT AND SECTOR INFORMATION

All Sectors used by RODOS & AMSDOS are 512 bytes long.
The following information applies to the RODOS format ONLY.

The sectors are numbered from &81 to &8A on the 1st side of the disc,
If a second side is also used, these are numbered from &8B to &94.

A sector can store 4 different types of information:

1 .. Root Sector Information block
2 .. Directory
3 .. Directory Continuation
4 .. File

The ROOT SECTOR BLOCK - is a special sector holding information about the disc (size, etc.). It is 512 bytes long, of which the last 256 bytes are a directory, This directory is the root directory that is displayed the first time a disc is DIRed.

The first 256 bytes are made up as follows:

Byte 0	number of tracks -1 (i.e. highest track number used)
1	number of sides -1 (0=single sided, 1=double sided)
2-17	Disc name in ASCII characters
18-255	Disc Allocation Block (see below)
256-511	Root directory

The Disc Allocation Block is used to determine what sectors on the disc are used, and which are free. This block is considered as a serial bit map (starting at bit 0 of byte 0, and then going through bits 1,2,3,4,5,6 & 7, followed by byte 1 bit 0, and so on), where each bit represents 1 sector on the disc.

The number of bytes used in this block depends on the number of tracks and the number of sides formatted on the disc.

A bit of 1 shows that the sector is in use.
A bit of 0 shows the sector is free to be used.

The first bit of this block is always set as it represents the first sector on the disc (track 0, sector &81), which is the root sector block itself.

Sectors are numbered from &81 to &8A (continuing with &8B to &94 on a double sided disc), tracks are numbered from 0.

The FILE LAYOUT.
The file Is broken down into 508 byte chunks when stored in sectors, with additional 4 bytes of further information.
The 512 byte sector is laid out as follows

0 - track of last 512 bytes (or dir track if start)
1 - sector of last 512 bytes (or dir sector)
2 - track number of next 512 bytes *
3 - sector number of next 512 bytes *
4 - 511 file bytes

* These bytes hold the length of the data in this block if it is the last block. Byte 2 = low 8 bits, Byte 3 = bit 9 of length.

The DIRECTORY LAYOUT

The directory is made up of three sections, which are :

- The directory header
- The directory entries
- End or continuation marker

The Directory Header is 4 bytes long:

Byte 0 - "D"
1 - "R" (or "r" on a directory continuation sector)
2 - track number of father dir
3 - sector number of father dir

Bytes 0 and 1 are used to check that the directory is really a directory, and if not an error is reported. Bytes 2 and 3 hold the track and sector of the directory that this one is in (or the previous track and-sector if this is a continuation).

The Directory Entries are each 32 bytes long:

0 -	Directory access byte
1-16	Filename or Directory name
17	File Access.byte
18	track of start of file
19	sector number of start of file
20	File type
21-22	File length
23-24	Location
25-26	Entry address
27-28	Owner number
29	Link Drive number
30-31	Reserved for future expansion

The Directory Access Byte - a '1' or a '2' indicates an end marker, otherwise this is a bit significant byte:

bits 0/1 - end marker detect
bit 2 - if set, this entry is for a sub directory
bits 3-5 - not used
bit 6 - display name flag (0=filename is displayed on a !DIR)
bit 7 - file exists (0=file exists, 1=erased)

The File Access Byte is also bit significant, and can be thought of as two groups of information.

Bits 0,1,2 account for the owner of the file (see Owner Number)
Bits 3,4,5 account for all other user numbers
bit 0 & 3 - Read permission (1=Read OK)
bit 1 & 4 - Write permission (1=Write OK)
bit 2 & 5 - Execute flag (0=Execute)
bits 6 & 7 are not used.

The Owner Number is a 16 bit user ID.

The Link Drive Number indicates if file is on a different drive 254=no link, 255=same disc, 0..253=new drive number (see !LINK).

The END MARKER

This byte follows the last directory entry in each sector.
If this byte is a 1 - the following 2 bytes hold a track and sector of where the directory continues.
If this byte is a 2 - the directory has finished.

****

APPENDIX D:
MEMORY BUFFERS

Any opened file (INPUT/OUTPUT/RANDOM) causes & header record to be created in memory. This header record is outlined below, along with an offset from the address returned for the header by the firmware open-file routines.

-15	Buffer type ("O" "I" or "R" out/in/random)
-13/14	Length free in buffer
-12	Drive number
-11	Dir track
-10	Dir sector
-9/8	Length used in this sector
-7	File access byte
-5/6	Buffer pointer
-3/4	Amount left to read in this buffer
-2	Current track
-1	Current sector
0-15	Name of file
16/17	Not used
18	File type *
19/20	Length of file *
21/22	Location of file *
23	Not used
24/25	Logical length *
26/27	Entry address *
28	Previous track -- start of 512 byte sector buffer --
29	Previous sector
30	Next track
31	Next sector
32/539	File data store

Bytes -15/-13 and 28/539 are added when the file is first accessed, and go into the buffer passed in DE during the open-a-file firmware call.

****

The PRINTER BUFFER

This works on a wrap around buffer system, i.e. data is added at one end, and removed from the other, causing the buffer to move up through memory. When the buffer reaches its top limit, it will return to the lower limit, and thus wrapping around. If the buffer is active, RODOS will check 50 times a second to see if anything is waiting to be sent to the printer, and if so, it will send it (if the printer is ready).

This buffer can be anywhere in RAM (incl. any expansion RAM supported) but only up to 64k in size. Any char sent to &BD2B will be added to the buffer.
(PRBUFF calls MC WAIT PRINTER &BDF1 to see if the printer is in use, and MC SEND PRINTER &BD31 to send the characters).

****

APPENDIX E:
The ERROR NUMBERS and MESSAGES

When RODOS reports an error, it puts an error number into &BE6D, so that you can check what error (if any) was reported.

An example of its use would be:

```
POKE &BE6D,255 clear error
!MKDIR, "newdir"
IF PEEK(&BE6D)=13 THEN print"newdir already existed":END
```

The error numbers and messages are as follows

1 Too Many Parameters
2 Bad File Name
3 Wrong number of parameters
4 Bad dir
5 Bad character
6 Unknown command
7 Access denied
8 ... not found
9 No match (not found with wildcard)
10 Disc full
11 AMSDOS ? (cannot find AMSDOS)
12 CPM ROM missing (checked during startup)
13 Dir already exists
14 . Bad drive
15 Unknown file system
16 Input file not open
17 Output file already open
18 ... Reborn (file not expected to exist)
19 ... Already exists
20 Bad format specified
21 Corrupted disc error
22 Disc not formatted
23 Bad file
24 Directory not empty
25 Can't link to a linked file
26 Bad alias defined
27 To many aliases
28 Disc read error
29 Disc Write error
30 Disc tracking error
31 Disc missing
32 Disc fault
33 Disc write protected
34 Title too long *(Note: V2.22 only)*

****

APPENDIX F:
PATCHING FOR USE WITH HARD DISCS OR OTHER RAMS FOR SILICON DISC

To set up RODOS for use with hard discs of silicon discs, some knowledge of machine code is required as some RSX's must be set up.
First a few ROM addresses must be looked up (Commands !^D !^E and !^F) and then stored. RSX's must then be set up for these 3 commands.
Each of these RSXs must check the given drive number against its own (see Appendix A), and then, if it 1s not the required drive, a jump should be made to the ROM address of the original command.

Registers as used when entering ^D (Read sector from disc)
HL = address of 512 byte buffer to put/get bytes
E = drive number 0-254 D = track number C = Sector number

Registers as used when entering ^E (Write sector to disc)
HL = address of 512 byte buffer
E = drive number D = track number C = sector number

Registers as used when entering ^F (Format a track)
HL = track data area E = drive number D = track number

The track data area is made up of four-byte-entries: trk/hd/sc/sz (one entry per sector to be written)

byte 0 - track number
1 - head number 0/1
2 - sector number
3 - size of sector (size=2^(byte+7) - i.e. 2 -> 512 bytes) followed by 0/1/2/3 for next sector, etc.

****

APPENDIX G:
THE VARIABLE WORKSPACE AREA

Each ROM has its own workspace area, which does NOT have a set place in the RAM. This workspace is for the ROMs use only (such as to hold buffers etc.), and only the ROM knows where it is. RODOS's workspace is about 2.5K most of which is taken up by a 2K file buffer for use in such things as !CAT, etc. Any entries that may need to be changed can be changed by the use of !OPT. There are a few, however, that good m/code programmers may be able to use, namely:

Offset	Size	Use
0	1	RODOS rom number
1	1	CPM rom number
3	1	Current drive letter
4	1	Current drive number
11	1	Expansion Ram count (in 16k blocks)
12	1	Start prbuff bank
18	1	Current side of disc (0 or 1)
23	2	Address of input buffer (file header)
25	2	Address of input buffer (sector block)
27	2	Address of output buffer (header)
29	2	Address of output buffer (sector block)
66	1	Home drive number for !CD
67	1	Home drive letter for !CD
68	1	Home track for !CD
69	1	Home sector for !CD
78	2	RODOS user number
88	2	I/O Port for expansion drives.

****

APPENDIX H:
USING RODOS FROM MACHINE CODE

To use RODOS commands from machine code, use KL FIND COMMAND to look up the name of the routine, and call it with A holding the number of parameters, and IX holding the parameter list. Note that different ROM versions have different entry addresses, so no assumption should be made about the addresses staying the same. Normal calls to firmware addresses patched by RODOS will work without any change and most m/code programs should work without any modification.

****

COMMAND INDEX

APPENDIX