Crunch-Mania V1.91 User’s Guide

COLLABORATORS

	TITLE : Crunch-Mania V1.91 User's Guide		
ACTION	*NAME*	*DATE*	*SIGNATURE*
WRITTEN BY		August 2, 2022	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

Chapter 1

Crunch-Mania V1.91 User's Guide

1.1 Crunch-Mania V1.91 User's Guide

```
                                 Crunch-Mania User's Guide

                Registered / Test Version 1.91 - 11.02.94


                        Written by Thomas Schwarz

                 Copyright (c) 1991-94 by Thomas Schwarz
                           All rights reserved
```

1.2 Introduction

1.3 Copyright & Registration

Crunch-Mania is Copyright (c) 1989-94 by Thomas Schwarz. It was released under the concept of "Shareware". All versions until V1.6 are Public Domain, but starting with V1.7 they are Shareware. I have released a test version that may be spread to give you a hint on the program's abilities. If, after a testing period of some weeks, you find this program useful, please register. By cooperating with this concept you help to ensure continued development of this product.

Registered Users will receive the latest release version of Crunch-Mania, a faster library, the latest version of RTDD, includes and autodocs for the CrM.library and separate decrunch sourcecodes (in Assembler). The registered version has no annoying registration requesters and has some more features which are ghosted in the test version.

No person(s) or businesses other than the author are authorized to accept any registration or distribution fees in any form whatsoever, except as specified by the author. Only people who have sent US 20, DM 30 or more to the following address will be regarded as registered users. Only these two currencies and only bank-notes will be accepted.

Thomas Schwarz
Dammstrasse 12
70806 Kornwestheim
Germany

Just put the money in an envelope together with some piece of paper with your request and your address on it or print out the orderform and then post it to me.

Delivery may take two to eight weeks. Usually you will be sent your registered version right away, but if a new version is imminent to be completed, you will get that one as soon as it becomes available.

I am still searching for somebody or a company who distributes this program as a commercial product, so I can spend my time in developing new features and not in pasting stamps on envelopes. So be free to contact me to relieve my pain!

The reqtools.library this program uses starting with V1.8 is copyright 1991/1992 by Nico François. That's what I call cool!

1.4 Distribution

Basically, the test version of Crunch-Mania may be distributed freely as long as the following conditions are met:

The Distributor may only charge a fee up to the costs of obtaining any common public domain disk. The distributor agrees to stop distributing the programs and data involved on request of the author. The distributor may only distribute an unmodified copy of the original program, with all the supplied documentation and copyright notices left intact.

1.5 Support

If you have any suggestions, bug reports or questions, please contact the author at the above address.

When sending bug reports, please describe exactly under what circumstances the bug occured, what equipment was used and what happened. If possible also try to give me enough information to reproduce the bug. It is very difficult to find bugs when you don't know exactly what happened. Please don't just send messages like "my machine sometimes crashes when I try to load xxx", that really doesn't help me. If possible, submit the offending file in crunched and original state to me so I can test it myself, or give me a pointer where I can find the files.

1.6 License

1. This license applies to the product called "Crunch-Mania", a set of programs for the Commodore-Amiga computer, published by Thomas Schwarz under the concepts of ShareWare, and the accompanying documentation, example files and anything else that comes with the original distribution. The terms "Programs", "Crunch-Mania" and "CrM" below, refer to this product. The licensee is addressed as "you".

2. You may copy and distribute verbatim copies of the programs' executable code and documentation as you receive it, in any medium, provided that you conspicuously and appropriately publish only the original, unmodified programs, with all copyright notices and disclaimers of warranty intact and including all the accompanying documentation, example files and anything else that came with the original.

3. You may not copy and/or distribute these programs without the accompanying documentation and other additional files that came with the original. You may not copy and/or distribute modified versions of these programs.

4. You may not copy, modify, sublicense, distribute or transfer the programs except as expressly provided under this license. Any attempt otherwise to copy, modify, sublicense, distribute or transfer the programs is void, and will automatically terminate your rights to use the programs

under this license. However, parties who have received copies, or rights to use copies, from you under this license will not have their licenses terminated so long as such parties remain in full compliance.

5. By copying, distributing and/or using the programs you indicate your acceptance of this license to do so, and all its terms and conditions.

6. Each time you redistribute the programs, the recipient automatically receives a license from the original licensor to copy, distribute and/or use the programs subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein.

7. You may not disassemble, decompile, re-source or otherwise reverse engineer the programs.

8. You may use the programs for a period of up to 30 days for testing purposes. After that, you have to register.

9. If you wish to incorporate parts of the programs into other programs, write to the author to ask for permission.

10. You agree to cease distributing the programs and data involved if requested to do so by author.

11. You may charge a fee to recover distribution costs. The fee for diskette distribution may not be more than the cost to obtain any common public domain disk.

12. Only the test version is intended to be distributed, the registered version may only be used by registered users and nobody else.

1.7 Disclaimer

THERE IS NO WARRANTY FOR THE PROGRAMS, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAMS "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAMS IS WITH YOU. SHOULD THE PROGRAMS PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY REDISTRIBUTE THE PROGRAMS AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAMS (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAMS TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

1.8 Future Plans

The program is planned to be a real multi-talent. So a disk archiver and a files archiver will be added as soon as I have any spare time. But my special attention will be given to the development of faster and more effective algorithms. Another goal is to adapt the GUI to Commodore's styleguide rules.

If you register and support my program, you will speed up the release of new and better versions. If there is little demand I may drop this project and start coding something else and that would be a great pity.

1.9 Why buy the registered Version?

Well, first of all you should buy the registered version, if you use this program for a longer time. With a registration you will support future releases. The registered version is faster both in crunching and in decrunching. With the registered version you can decrunch reloc files and not only data files. Registered users get the latest version of my run-time-data-decruncher (RTDD) which is very useful e.g. for crunching icons or include files, they will be automatically decrunched. The source code of the decruncher is also provided to be implemented in your own releases. Registered users receive a library that crunches and decrunches faster. A registered user gets the latest version of this product plus the next version as soon as it gets available! A program worth using is a program worth buying!

1.10 General Information

1.11 Outward Appearance

The program opens a 2 planes intuition screen which is *NOT* a public screen. It uses the intuition.library for full multitasking compatiblity, the gadtools.library for nicer gadgets and menus and the reqtools.library ((c) Nico Francois) for nice requesters. It is self-detaching so you do not need to call it with "run". You can either run it from CLI or from the Workbench.

The program is controlled by gadgets and menus. On the right side there is an area for tables and error messages and at the bottom there is a statusline combined with a percentage display which indicates what the program is just doing.

1.12 Numbers Input & Output

Input:
You can enter hex and decimal numbers, but hexadecimal numbers are only recognized with a leading "$". To get negative numbers put a "-" at the beginning of the line. An empty gadget has the same effect as one with a zero, sometimes this also means that you want to disable this option, e.g. an empty "Stack At"-gadget means that you don't want to set the stack to a special address. Some extra functions can be reached by entering chars instead of numbers.

Output:
The output follows the same rules as the input: a "$" indicates a hexadecimal number, the rest is decimal. But there is one exception: while loading executables in the Address- or Link-Mode the dehunker displays hexadecimal numbers without a leading "$". This has not been forgotten but left out as a result of lack of space.

1.13 System Requirements

For packing small files 512KB may suffice but the full power of this program can only be obtained with at least 1MB memory resulting from 200-400KB of tables depending on the version, the crunchmode and the offset. Two drives are comfortable but not necessary. No problems have occured under Kickstart 1.2, 2.0 and 3.0, so it should also work under 1.3, 2.x and 3.x+.

1.14 Other Information

There are no known bugs. Nevertheless the author is in no way liable for any errors and gurus invoked by the Program. In most cases the user himself produces the errors by misusing the Program.

This document will only deal with the important things. Understanding minor details like Filelen is expected. These things are left out.

1.15 A Word on the Keys

You may invoke actions by pressing special keys. These actions are the same as the corresponding gadgets would cause. The key that stands for a certain gadget is underlined in the text next to or in the gadget.

When pressing a key, the gadget behaves as if you had used the mouse, i.e. it looks like being pressed down. In this state no actions have yet happened, you may cancel your choice by pressing one of the <Shift> keys. Otherwise the action will be performed as supposed. Only one key at the same time is regarded as pressed, keys that are pressed while an other key is still pressed will be ignored.

1.16 CLI-Options, Menus & Gadgets

Command Line Options

Menu Options

Gadgets

1.17 Command Line Options

Here is a list of all options and parameters available and ↩ adjustable
via the CLI. The same options can be used in the configuration file "S:CrM.cfg". This file is processed before the command line, so you can change your defaults via the CLI. A "+" instead of a "-" in front of the option reverses the result.

<?> - CLI Help

<-c> - Set Screen Colors

<-d> - Enable Auto Makedir

<-e> - Set Extra Mem Size

<-f> - Enable LED-Flash

<-g> - Enable Segment-BPTRs

<-h> - Set Huffman Buffer Size

<-i> - Set Subdir-Level

<-I> - Ignore Configfile

```
<-p> - Set Taskpriority

<-r> - Enable reqtools.library

<-s> - Enable Sample Encoding

<-t> - Set Textbuffersize

<-u> - Enable Free Mem Updating

<-y> - Enable Data Encryption
```

1.18 <?> - CLI Help

This option lists all currently available options in the CLI to give you a brief overview of what you can do there and then asks for a new command line. This option has to be the first in the command line and no other options may be specified with it.

EXAMPLE

```
1> CrM ?

Crunch-Mania V1.9r - Copyright (c) 1991-93 Thomas Schwarz

Command Line Options:
       .
       :

Enter Command Line:
```

(Well, with <Esc>-codes this looks a bit nicer)

1.19 <-c> - Set Screen Colors

With this option followed by four hexadecimal numbers you can set your own personally preferred screen colors fitting your needs. The highest nibble of each number stands for the red part of the color, the mid nibble for the green part and the lowest nibble for the blue part.

EXAMPLES

```
1> CrM -c$889 $2 $ccd $358               ;These are the default colors.
or
1> CrM -c$0 $bbb $2a2 $137
```

1.20 <-d> - Enable Auto Makedir

This option comes into effect when you crunch whole directories or use the multiselect feature of the filerequester and you select a subdir. When this option is active this dir will be automatically created when CrM tries to access a file located in this dir, otherwise there will appear an information requester. When you deny this requester the whole dir is skipped.

EXAMPLES

```
1> CrM +d                              ;lets a requester appear
```

1.21 <-e> - Set Extra Mem Size

As this program does not support floating compression, the source and destination memory blocks overlap to reduce the memory needed. Extra Mem specifies the distance between the two blocks.

The error message "File not crunchable!!!" indicates that the cruncher would have overwritten some data which has not yet been processed with crunched data. Nevertheless you can crunch those files by setting extra mem to a higher value, but the files usually then become longer than the original.

Extra Mem may be set by the option <-e> directly followed by a number in the range from 100 to 10000. Default is 100.

EXAMPLES

```
1> CrM -e100                           ; Sets Extra Mem to 100
or
1> CrM -e$200                          ; Sets Exrta Mem to 512
```

1.22 <-f> - Enable LED-Flash

While packing the power-LED flashes to show the user that the program is working. As you certainly know, the power-LED also steers the low-pass filter for the four audio channels. This may lead to an annoying sound when listening to a tune while crunching. To prevent this you can disable the flickering with the option <+f>, default is on.

EXAMPLE

```
1> CrM -f                    ;Enable LED-Flash (default)
```

1.23 <-g> - Enable Segment-BPTRs

This option only comes into effect in the address and link modes. There you can relocate executable to a certain address. The dos.library does the same but it adds the lenght of a hunk and the pointer to the next hunk in memory to all hunks. Normally this is only for internal DOS use, but some programs use these BPTRs to calculate the addresses of their hunks. You can enable the creation of BPTRs with this option so that these programs don't crash any more. As you don't need this option very often the default setting is off.

EXAMPLE

```
1> CrM -g      ;now BPTRs are created
```

1.24 <-h> - Set Huffman Buffer Size

This option can be used to set the size of the buffer used in the LZ-Huffman compression mode for collecting statistics. The size of this buffer affects the compression ratio in unpredictable ways. As a general rule, keep this at the default, but if you are compressing homogenous data with an approximately fixed relative frequency of symbols (like text files), setting this to a large value will improve compression. Binaries generally compress best with the default setting.

The Huffman buffer may be of any size between 4K and 64K and must be specified immediately following the <-h> string, in kilobytes.

EXAMPLE

```
1> CrM -h32                ;Sets the Huffman buffer size to 32768 (32K)
                           ;bytes
```

The default Huffman buffer size is 16K.

1.25 <-i> - Set Subdir-Level

CrM takes care of this option when you want to crunch an entire directory or use the multiselect feature of the filerequester and select dirs. This number specifies the level upto which the dirs are scanned. The number follows immediately after the "-i", see examples for more details. Accepted values range from "-1" to "127".

Let's say you want to crunch the whole ramdisk and let's say yours looks like mine below. If you specify "-1" as number the whole ramdisk will be compressed. If you enter "0" only "dirlist" will be crunched because it is the only file in the rootdir. When you take "1" as the number CrM will crunch "dirlist" and the files "ASM-One.Pref" to "ZOOMOPTS" in the env directory. Taking "2" as number will have the same effect as "-1" (in this example case).

```
RAM DISK:
     ENV (dir)
          GadToolsBox (dir)
            GadToolsBox.prefs
          egs (dir)
            mouse.prefs
          Sys (dir)
            floppy.prefs                     font.prefs
            icontrol.prefs                   input.prefs
            locale.prefs                     overscan.prefs
            palette.ilbm                     pointer.ilbm
            printer.prefs                    printergfx.prefs
            printerps.prefs                  screenmode.prefs
            serial.prefs                     sound.prefs
            sysfont.prefs                    wb.pat
            wbconfig.prefs                   wbfont.prefs
            win.pat
       ASM-One.Pref                     NEWLIST_LANG
       TERMPATH                         xprascii
       xprzspeed                        ZedZapWindow
       ZOOMOPTS
     Clipboards (dir)
     T (dir)
  dirlist

EXAMPLES

1> CrM -i-1                   ;take whole directory

1> CrM -i1                    ;crunch sourcedir and the files of the dirs
                              ;in it

Default is -1 (scan entire directory).
```

1.26 <-I> - Ignore Configfile

```
   In the configuration file "S:CrM.cfg" you may specify your own personal
default  settings  which  are  loaded  immediately  after the start of the
program.   This  option prevents CrM from loading the config file what may
come  in  handy  when you don't have your boot disk inserted.  This option
has to be the first in the command line but others may also be set there.

EXAMPLE

1> CrM -I +r         ;don't load config file and don't load reqtools.library

1> CrM -I            ;don't load config file and use internal defaults
                     ;instead
```

1.27 <-p> - Set Taskpriority

This value sets the taskpriority of the background task which is responsible for loading, crunching and saving. The task handling the windows, menus and gadgets always has a higher priority so that it is not frozen while crunching. Nevertheless you have to select the priority wisely because high priorities will interfere multitasking which makes this machine so powerful! This value ranges from "-128" to "126".

EXAMPLE

```
1> CrM -p-5                            ;Set Pri to -5
```

Default is -1.

1.28 <-r> - Enable reqtools.library

The reqtools.library is loaded by default and is used for user communication. When you are short of memory you may prevent CrM from loading it with the option <+r>.

The reqtools.library is copyright Nico Fracois. It is used for filerequesters, numberrequesters and informationrequesters. CrM can operate without this library, but it is more comfortable to use it.

```
1> CrM +r            ;don't open reqtools.library
```

1.29 <-s> - Enable Sample Encoding

This feature is especially designed for compressing 8-Bit samples. Generally this option will only reduce the crunched size with samples as this algorithm is comparable with delta-encryption. It calculates the relative difference between two bytes instead of the absolute values. Now imagine a sine wave. The difference between two bytes is in the range of -4 to +4, but the absolute values go from -128 to +127. The algorithm will convert the file into one with about 20 different values instead of 256.

To give you a hint on the efficiency of this algorithm I have put up the following chart. The files are standard 31 Instruments Protracker modules taken from the "Mayday Attack Vol. II" packed with the LZ-Huffman compression algorithm. The proportional numbers reflect the saved bytes.

Name	Original	Sample-Mode=Off		Sample-Mode=On	
mod.acid_yell	230764	19.7%	185338	27.0%	168350
mod.badada	112742	35.1%	73196	41.2%	66338
mod.flatliner	140886	29.3%	96648	47.8%	73612
mod.fuck_you!	142652	24.8%	107238	33.3%	95202
mod.gotta_speed	108594	21.4%	85338	33.7%	71992
mod.hard_attack	108100	47.0%	57260	54.6%	49118

```
  mod.leck_mich_am_arsch |   125486 | 39.0%     76528 | 46.3%     67338
  mod.something_big      |   162756 | 19.5%    131092 | 27.2%    118556
  mod.world_of_terror    |   102114 | 48.2%     52932 | 51.8%     49174
  mod.you_gotta_believe  |    85092 | 24.7%     64084 | 26.6%     62490
  -----------------------+----------+-----------------+----------------
  Total                  |  1319186 | 29.5%    929654 | 37.7%    822170
```

EXAMPLE

```
1> CrM -m    ;Enable Sample Encoding
```

Default is off, as most files compress worse with it.

1.30 <-t> - Set Textbuffersize

This value specifies the size of the history buffer in lines. On Kick1.2/1.3 this value is fixed to 18, as my version of the gadtools13.library always crashed when I tried to scroll the outputfield whith the slider next to it. You may enter values from "18" to "999".

EXAMPLE

```
1> CrM -t200                          ;Set history size to 200 lines
```

Default is 100.

1.31 <-u> - Enable Free Mem Updating

The title bar of the screen or the window displays the currently available unused memory. This display is updated every 1/10 second. You can disable this with <+u> to gain a bit more speed, but it's not very much.

EXAMPLE

```
1> CrM +u                   ;no updating
```

1.32 <-y> - Enable Data Encryption

Starting with CrM V1.9 you have the possiblity to encrypt your data with a password. You have to enter this password when the crunching is finished. You will see only spaces in the requester, therefore you have to enter it twice to be sure that you have entered it correctly. Currently this only works in the Data-Mode, in all other modes the state of this flag is ignored.

EXAPMLE

```
1> CrM -y                              ;Enable pw encryption

Default is off.
```

1.33 Menu Options

```
                   In the following lines the functions of the menus are  ↩
                      described.  Menus
are accessible via the right mousebutton.


                Project Menu

                About     - <a>

                Load File - <l>

                Crunch    - <c>

                Save File - <s>

                Save Prefs

                Iconify   - <i>

                Quit      - <Q>

                Options Menu

                Free Mem Update

                reqtools.library

                Segment-Ptrs

                LED-Flash

                Sample-Mode

                Encrypt Data

                Auto Makedir

                Priority

                Include Subdirs

                Extra Mem

                Huffman-Buf
```

Text Buffer

1.34 Project Menu

In this menu the basic operations have found their place. They have the same effect as the corresponding gadgets, just for people who like menus more than gadgets. For "About" and "Save Prefs" there is no corresponding gadget.

1.35 About - <a>

This will display the text that is shown when you start the program. When the reqtools.library is active a requester is called instead. In the registered version this requester also displays the user's address.

1.36 Load File - <l>

Load a File, has the same function as Load gadget, see
Load
.

1.37 Crunch - <c>

Crunch the loaded data, has the same function as Crunch gadget, see

Crunch
.

1.38 Save File - <s>

Save a File, has the same function as Save gadget, see
Save
.

1.39 Save Prefs

This function has no equivalent gadget. It creates the config file "S:CrM.cfg" and saves the current status of the menus to it. You may edit the created file with any texteditor for the prefs are saved as normal ASCII-text.

1.40 Iconify - <i>

Close screen and window and open a small window on the ↩
workbench
screen, same effect as Iconify Gadget, see
Iconify
.

1.41 Quit - <Q>

Leave Crunch-Mania, same as Quit gadet, see
Quit
.

NOTE: you have to press <Shift> when using the keyboard shortcut.

1.42 Options Menu

This menu contains several preferences and values that are used in the program. There are no corresponding gadgets, so you have to use the menu. Some options are disabled when the the reqtools.library is not active.

1.43 Free Mem Update

See
<-u> - Enable Free Mem Updating
for details. Default is on.

1.44 reqtools.library

See
<-r> - Enable reqtools.library
for details. Default is on.

1.45 Segment-Ptrs

Consult
<-g> - Enable Segment-BPTRs
for more information. Default is
off.

1.46 LED-Flash

See
<-f> - Enable LED-Flash
for further information. Default is on.

1.47 Sample-Mode

See
<-s> - Enable Sample Encoding
for more details. Default is off.

1.48 Encrypt Data

See
<-y> - Enable Data Encryption
for datails. Default is off.

1.49 Auto Makedir

See <-d> - Enable Auto Makedir for details. Default is on.

1.50 Priority

Here you can set the task priority of Crunch-Mania. There are three given possibilities (-3, 0, 3) the custom option The current task priority is shown behind the double point. Select your priority wisely, because high priorities will interfere multitasking, and that's what makes this machine so powerful! See <-p> - Set Taskpriority for more information. Default is -1.

1.51 Include Subdirs

Here you can set the rekursion depth for scanning dirs. There are two predefined possibilities, "All Subdirs" which equals -1 and "No Subdirs" meaning 0, and the Set Level option where you can set any level between -1 and 128. See <-i> - Set Subdir-Level for more details. Default is -1.

1.52 Extra Mem

See <-e> - Set Extra Mem Size for a detailed description. The current value is shown after the double point. You may change it by selecting the menu item. The new value will be used when the next file is loaded. Default is 100.

1.53 Huffman-Buf

See
<-h> - Set Huffman Buffer Size
for more details. Default size is
16K.

1.54 Text Buffer

See
<-t> - Set Textbuffersize
for more details. Default size is 100
lines.

1.55 Gadgets

Most gadgets behave as you expect it. Only the mode ↩
gadgets
(Reloc/Data/ Address/Link) are a bit special. You can only change the
mode when the buffer is empty. To empty the buffer you have to press the
abort gadget.

Quit

Abort

Iconify

Crunchmode

LowMem & HighMem

Load|Fix File to

Load

Algorithm

Offset

Crunch

Decruncher

Decrunch to

Jump to

Stack at

Decruncher at

Stop Drives

SR (Status Reg)

Move Gadgets

Save

LZH-Table At

Advanced Options

Slider Bar

1.56 Quit

Uuh, I forgot the function of this gadget. Well, I think you should not use it, cause it's the end of everything. It's the most senseless gadget of all. Anyway, when pressing it a requester is invoked where you have to confirm your choice.

1.57 Abort

Generally you need this gadget only to stop crunching. But you also have to press this button when you e.g. loaded something in reloc mode and now want to load something in data mode. You will experience that you can't do this unless you press this gadget. You may not change the crunchmode when something is loaded because then the decrunchers would not fit. By pressing this gadget you unlock that feature.

1.58 Iconify

This closes the main screen and opens a new small window on the workbench screen. The new window displays the togo-counter, so you can crunch something and see when it has finished. You may return to the main screen by pressing the right mousebutton when the window is active.

1.59 Crunchmode

Here you may select between the four crunchmodes. These are: Reloc, Data, Address, Link. There is also an "Auto" Checkbox. When it is selected CrM automatically detects whether a file is executable or not and changes the Crunchmode accordingly. This is only for registered users.

In the reloc mode you can load any executable except overlayed files (I think there are not too many). The hunk structure is analysed and displayed while loading. Name, Symbol and Debug hunks are skipped, Reloc hunks are sorted and precompensated. This new reloc table format is shorter and even better crunchable than the one DOS uses!

The data mode was designed for normal not-executable data files. They are loaded into memory, compressed and then saved with a small data header. The small comment which is also saved is for RTDD which takes the original len out of it without opening the file resulting in a faster access when reading the dir.

The address mode is a bit similar to the ByteKiller cruncher. Here you can load executables and fix them to a special address, load position-depentant code or just load pure data. The decruncher then puts this data to a specified memory block and jumps somewhere. This crunchmode is not very DOS-friendly (especially the decruncher) but very handy e.g. for bootloading games. The dehunker displays some hexadecimal numbers without a leading "$", so don't get confused.

The link mode resembles tetrapack/defpack. In this mode you can load several executable and data files into a precleared memory block which you have to define first. Executables are relocated to certain addresses inside the block, data files are loaded to a specified position in the block.

I recomend to use the last two modes only with proggies which do not return to the CLI because important system-parameters and -tables may be destroyed.

When the Auto checkbox is checked CrM behaves in an itelligent way. It detects whether the loaded file is executable or not, and then decides which mode (reloc or data) is the right one.

1.60 LowMem & HighMem

These two gadgets are only available in link mode. They specify the memory block to which the files are loaded. This block is buffered someplace else in memory while loading and crunching, so you don't have to care about crashes. LowMem and HighMem both have to be even addresses and HighMem has to be higher than LowMem. It is recommended to set the "Decrunch to" address to LowMem to avoid unwanted results.

1.61 Load|Fix File to

These two gadgets are active in the address and link mode. The "Load / Fix" gadget determines whether an executable is loaded as a data file (Load) or relocated to the address given in the string gadget below it (Fix). For data files this cycle gadget has no effect. In the address mode the string gadget is only important for executables, it contains the address where to fix them to. In the link mode this gadget affects both executable and data files in the same way: with the address in the gadget an offset to the beginning of the memory block is calculated and the file is then loaded there. The file has to fit completely in the given memory block.

1.62 Load

With the "Load" gadget you may load files in all modes. When there is no filename given in the string gadget next to it, or when the filename in the string gadget has not been changed since the last loading, a reqtools.library filerequester apppears (when enabled). After pressing "Load" and then selecting a file the program immediatly begins to load the file. You may select more that one file and even dirs by holding down the <Shift> key. Then you will be asked for a destination dir.

With the DIR gadget next to the load filename you may select a directory to crunch. Now the whole directory will be packed, subdirs may be included or not, this depends on the Include Subdirs Level (see chapter 3.1 for details). You may activate the string gadget by pressing <Return>.

When loading an executable its hunk structure is displayed in the output field either with decimal numbers (reloc) or with hexadecimal numbers (adress and link), where Range is the part the entire hunk occupies and Len is the length of the actual data in the hunk. The filelength is shown below the load string gadget. "Processed Len" only appears in the Reloc mode in the display field and shows the size of the stripped and precompressed data. This is the len of the data the cruncher really crunches.

When you load a file that has been crunched in reloc or in data mode it is automatically decrunched presuming you selected the right mode (reloc for executable files and data for data files) so you can recrunch it or save the original file. After it has been decrunched the file is treated as if you had loaded it in its original form.

1.63 Algorithm

This is again a rotating gadget. You may select beween these two crunch algorithms: LZ-Huffman and CrM-Normal.

LZ-Huffman is the best of these algorithms, but not the fastest. Files crunched in this mode need 1248 bytes of additional memory to decrunch. The decruncher is also slightly slower than the one of the CrM-Normal mode. To crunch files in this mode you need continuous memory blocks with the sizes:

```
   128K
 + 5*Crunchoffset
 + Huffman Buffer
```

With the default values of $7ffe and 16K this will lead to a memory usage of 304K.

The advantage of the CrM-Normal mode is its fast decruncher and its low memory usage. But its results are the worst of the two modes partly caused by the Crunchoffset which is limited to $4200. It needs no additional memory to decrunch. The memory used while crunching is shown in the following table:

```
   128K
 + 5*Crunchoffset
```

With the default Crunchoffset of $4200 this will lead to a memory usage of 210K.

1.64 Offset

This value specifies the range in which redundant data is searched and eliminated. Higher values produce better results but also longer crunch times. Values range from 1 to $4200 in the CrM-Normal mode and from 1 to $7ffe in the LZH mode. Values which are too high will be automatically reduced to the maximum.

1.65 Crunch

Now you have started the real crunching process. This will take a while, so be patient. The ToGo counter is displayed in the status line at the bottom. It shows how much data has not yet been processed. This counter is permanently updated and is also shown in the Iconify window. When finished the crunched len of the file is displayed above the save string gadget. This is not the final length! The decruncher or data header is not yet added.

The error "File not crunchable!" indicates that there is not enough space between the original and the crunched data. You may increase Extra Mem to avoid this message, but generally these files become longer when crunched.

The message "Min Security-Distance" shows the distance between the source and destination memory block which is needed to decrunch a file in the same memory block properly. The crunched data has to be in front of the decrunched data. For more information have a look at the decruncher source.

1.66 Decruncher

With this gadget you can select the decruncher you want. Each crunchmode has a special set of decrunchers. Not all gadgets that are displayed are also needed for the selected decruncher. So be careful to select the right decruncher especially in the address and link mode.

First of all there is the data mode. As it is really senseless to put a decruncher before a data file there is only the data option. This will save a 14 bytes data header in front of the encoded data which is necessary for decrunch. This header is described in the decruncher source. This decruncher does not any other input.

Then there is the reloc mode. If you choose the simple decruncher (it's the shortest one with implemented decruncher) you have a pretty low level decruncher. When a crunched program is running, twice as much memory is used as needed. The decruncher doesn't free the needless memory but lets this do the dos.library after the whole program has done its job. You can only use this decruncher in combination with the CrM-Normal algorithm. This decruncher takes no input from other gadgets.

The decruncher you should use is the normal decruncher. It frees the unnecessary memory before running the program. Here you can choose whether to kill the decruncher or not. Killing it will result in a bit more free memory, a longer decruncher and the cli_module in pr_CLI in the process structure pointing on the first hunk of the program (normally it points on the decruncher). Some programs need this (e.g. CygnusEd). As you might have guessed this decruncher pays attention to the "Kill Decr" gadget.

The latest decruncher is the library decruncher. It is the shortest one, but you need the CrM.library to run a program with that decruncher. With this decruncher you should have no problems with the cli_module pointer. You can use this decruncher to crunch libraries, fonts, devices, handlers (careful with the last two, crunch them only with Kick2.0+ because DOS uses an internal jumptable that cannot be patched in Kick1.2/1.3). This decruncher is independent of any gadgets.

The address and link modes have the same decrunchers. Normally it suffices to use the normal decruncher. You can tell him where to decrunch the data and where to jump. When you use the LZ-Huffman algorithm you also have to supply a value to the "LZH-Table at" gadget, because this decruncher needs some memory for a 1248 bytes large table to build the huffman tree.

The professional decruncher has some more options. You can set the stack as well as locate the decruncher somewhere in memory. You may stop the drives, set the status register and init some important hardware registers or addresses. You may let out some options by clearing their gadget. This packer saves only the really necessary parts and not an ultimate decuncher where only some values are changed and much senseless code is dragged along.

Finally you can save the files as data files. This has the same effect as just loading them in data mode.

1.67 Decrunch to

The "Decrunch To"-address specifies the address where the data is decrunched to. By entering here a "p" no special address is taken. Instead of this some mem is allocated (in the Hunk-Header) and there the data will be decrunched to. "c" has almost the same effect but here it will be in Chip-Mem (e.g. for proggies with internal copperlist).

1.68 Jump to

"Jump to" is the address where the decruncher will jump to after decrunching. In most cases this is the same address as the Decrunch-address. If there is a "p" or "c" as the Decrunch-Address, the Jump-Address is relative to the beginning of the data. By entering here a "r" the decruncher will return to the calling code after decrunching.

1.69 Stack at

This gadget specifies where the stack should be put. The Stackaddress is written into a7 (stackpointer) as the first action before decrunching.

1.70 Decruncher at

The decruncher can copy itself to a safe position which is fixed by "Decruncher at". Of course this is done before decrunching.

1.71 Stop Drives

You may stop all floppies either before decrunching or after decrunching (or both). The left column is responsible for all actions before decrunching and the right one for after. Stopping drives "lame" will switch off the motor but not the drive LED (at least it does so with my external drive). To turn it out you have to select "pro".

1.72 SR (Status Reg)

The value given in this gadget is written in the status register immediately before the program is entered. You may also enter values with the supervisor bit set (e.g. $2700), the status register is set with the help of trap #0. This may cause some troubles with accelerated machines when the vbr is not at address 0.

1.73 Move Gadgets

These are the twenty gadgets placed in five rows with two times two columns each. They are designed to write certain values to special addresses outa the data range. In the first column there are the values which can be wordened with a following ".w" (e.g. $0.w) and in the second one there are the addresses which are automatically shortened if possible (like in Ass: e.g. move.w #0,$dff180). The first two columns are executed before decrunching and the second two after. E.g. you can change the background color before decrunching to red and when decrunching has finished to black. Or you can turn off all interrupts and DMAs.

EXAMPLES

```
1) move  "$0.w     " , "$dff180  "            ;set colour0 to black
   move  "$f00.w   " , "$dff182  "            ;set colour1 to red
```

You can also combine the above two lines into a single one:

```
   move  "$f00     " , "$dff180  "            ;set colour0/1 to back/red

2) move  "$60000   " , "$6c      "            ;set level 3 interrupt to
                                              ;$60000
3) move  "$7fff.w  " , "$dff09a  "            ;disable all interrupts
```

1.74 Save

The save filerequester and filename follow the same rules as described in 3.3.7. The save directory may be different to the load directory. You can save the original file before pressing the crunch button or save the crunched file after crunching it. NOTE: decrunched executables are never totally equal to their original state, because some useless hunks are removed, reloc tables are sorted and Hunk_Ends are added. The final file lenght is displayed above the save string gadget.

When pressing the DIR gadget next to the save filename you may select a destination directory for multiple selected files and for crunching a whole directory. While crunching you can select a new save filename (Save gadget is ghosted).

1.75 LZH-Table At

This gadget determines where the LZH-Decruncher locates the tables it needs to decrunch the data. This gadget is only in the address and link modes important when you the LZ-Huffman crunch algorithm. The table is 1248 bytes large, so it won't be too hard to find some space for it.

1.76 Advanced Options

This gadget opens and closes the Advanced Options window. It contains the "Move" gadgets and "Stop Drives" gadgets. Compared to older versions it looks more jovial this way.

1.77 Slider Bar

The Slider Bar may be used to scroll the output field. When the slider bar is at the bottom each text that is printed out will cause the display to scroll up, if it is somewhere in the middle, the display will stay at it's place but the slider bar will scroll instead. When the slider bar reaches the top, the display will scroll again. This way you can reread something that was previously printed without being disturbed.

1.78 Supplied Utilities

This section describes the utilities which have to do with ↩ this packer
and which are supplied with this package.

Run-time Data-Decruncher (RTDD)

The CrM.library

The xpkCRM2.library

The xpkCRMS.library

CrMPW

CrMData

1.79 Run-time Data-Decruncher (RTDD)

This utility is very powerful and useful. It patches some (6 or 9) vectors of the dos.library, so any loaded datafile will be automatically decrunched. When the file is loaded entirely no extra mem is needed, otherwise the program will allocate a memoryblock as large as the original file and will decrunch it there. In all further read calls the wanted data is only copied from this buffer, so this will also speed up disk-access. In the current version (v1.9) RTDD supports all crunch algorithms (CrM!, CrM2), any Min Security-Len, the sample-mode and password encrypted data. In combination with the CrM.library you will be able to crunch almost EVERY file without realizing anything (except a little slow-down).

The following entries of the dos.library are patched:
- Open()
- Close()
- Read()
- Seek()
- Examine()
- ExNext()

When you are running Kick2.0+ these are also patched:
- ExAll()
- ExamineFH()
- OpenFromLock()

RTDD also supports some command line options:
"?" ··· This will display a help page.
"-i" ·· Install Only, DON'T remove RTDD. Useful for scripts.
"-r" ·· Remove Only, DON'T install RTDD. Useful for scripts.
When no options are given RTDD assumes that you want to change it's state from installed to removed or vice versa.

You CANNOT run RTDD from the workbench!

There have occured some strange effects when using RTDD in combination with Directory Opus. I will try to fix this in future versions.

RTDD V2.0

RTDD 2.0 has the same task as RTDD 1.9, but there is a major difference: it is a commodity. CLI options and icon tooltypes are the same and they will be explained together now.

```
CX_PRIORITY - Specifies the priority RTDD has among other commodities.
              NOTE: This is NOT the task priority!
CX_POPKEY   - Specifies the combination of keys which will let RTDD's
              window appear.
CX_POPUP    - Specifies whether RTDD's window will popup when RTDD is
              started.
PATCH       - Tells RTDD whether or not to patch the system.
CACHE       - Function not yet implemented
MAXMEM      - Function not yet implemented
```

EXAMPLE:

```
1> RTDD CX_PRIORITY=0 CX_POPKEY="ctrl alt d" CX_POPUP=NO PATCH=YES
        CACHE=NO MAXMEM=524288
```

The "Active" menu was designed for a very special case. Let's say you have installed RTDD and you want to remove it (for whatever reason), but you can't remove it, because another program has also patched the vectors that RTDD uses. With the old version there was no way to get rid of RTDD, but by turning off the "Active" menu you may prevent RTDD now from decrunching data while the patches are still installed.

1.80 The CrM.library

This program has the option to save crunched execuatables which will use the CrM.library to decrunch. Furthermore this library can be used by any programmer to decrunch data files easily. Each registered user will get the includes and autodocs of the library and may use it in his own productions. Registered users also get a faster version of the library.

Starting with release 1.9 (v4) the library also includes the crunch algorithms, so any programmer can use the library to crunch his (or her) own data. CrM will also use the library. Consult the autodocs for further information how to program it.

The library supports Sample encoded and password encoded files. When you want to decrunch a password crunched file the library sets up a reqtools requester where you can enter the password. The entered password is saved in a list so you don't have to enter the password all the time. When you enter the same password twice you may remove it (or all passwords) from the list.

1.81 The xpkCRM2.library

This library is an xpk interface for the CrM.library. It uses the crunch and decrunch routines of the LZ-Huffman mode the CrM.library provides. It behaves like any other standard xpk library, so there should not occur any problems with any application using xpk. It requires CrM.library v4+.

The internal benchmarks which are displayed with the xQuery command do not reflect the real performance of the library. As I don't own a A3000 i cannot execute them correctly. Sorry for the inconvenience.

1.82 The xpkCRMS.library

This library is an xpk interface for the CrM.library. It ↩ uses the crunch and decrunch routines of the LZ-Huffman mode in combination with the Sample mode the CrM.library provides. It behaves like any other standard xpk library, so there should not occur any problems with any application using xpk. It requires CrM.library v4+. This library is quite like the xpkCRM2.library, the only difference is that it uses additionally the Sample mode. Thus your data is also delta encrypted which may lead to better results with 8-bit samples. See

<-s> - Enable Sample Encoding
for more details.

The internal benchmarks which are displayed with the xQuery command do not reflect the real performance of the library. As I don't own a A3000 i cannot execute them correctly. Sorry for the inconvenience.

1.83 CrMPW

With this little program you can add or remove passwords to or from the list the CrM.library keeps to decrypt data. The use of this program is really self-explanatory, just type in the program's name to get an overview of the parameters.

The commands are case-insensitive, but the "all" has to be in lowcase to clear the whole list.

1.84 CrMData

This is another tiny but useful utility in the Crunch-Mania package. It allows to crunch and decrunch data files via the CLI. You may also use it in combination with your directory utility (e.g. DiskMaster, DirOpus). To get a list of the supported options just call CrMData without any parameters. The parameters have the same result as the corresponding switches in the main program (CrM).

1.85 Trivialities

Many Thanx may reach:
- Commodore for this fantastic Computer with Multitasking
- my HD for having no head-crashs anymore after trashing the v1.8 source
- Rene for Beta-Testing
- Michael Mutschler for Beta-Testing and detecting the Enforcer Hits
- Karsten Weiß for telling me that my GUI is not "styleguide"-like
  and for testing the xpk sublib (now it works perfect!)
- Alexander Ehlert for his Oberon Interface

Greetinx and Handsheix fly over to:

Michael, Rene, Tobias, Karsten, Christian, Christoph, Uli, ...

1.86 Performance Test

This test was done on my Amiga 500 with 7.14MHz, 512K Chip, 512K fake-Fast, 2M real-Fast, 105M HD, with all files being loaded from dh0:.

1) "PageStream2" - Main program of Pagestream V2.2, crunched as executable
437576 bytes, 110 Hunks, 22369 Reloc Offsets
Using Library-Decruncher when available, Times with loading from dh0:

Cruncher	Crunched Len	Crunch Time	Decrunch Time
CrM 1.8r LZH	178848	235s	12s
CrM 1.8t LZH	178848	352s	15s
CrM 1.8r Normal	186932	179s	10s
CrM 1.8t Normal	186932	303s	10s

2) same file as in (1), but crunched as data file

Cruncher	Crunched Len	Crunch Time	Decrunch Time
CrM 1.8r LZH	211058	285s	14s
CrM 1.8t LZH	211058	429s	18s
CrM 1.8r Normal	230344	185s	12s
CrM 1.8t Normal	230344	331s	12s

3) "Lha.man" - Text File (Doc of Lha 1.42), 122451 bytes

Cruncher	Crunched Len	Crunch Time	Decrunch Time
CrM 1.8r LZH	36872	55s	2s
CrM 1.8t LZH	36872	83s	3s
CrM 1.8r Normal	42648	46s	2s
CrM 1.8t Normal	42648	79s	2s

The speed and ratio comparisons towards other packers have been removed to ensure equal competition. I don't want to urge people to my packer, they have to find out on their own which packer is the best.

1.87 History

```
********** V1.3 **************
First Release Version, with almost all Options as V1.4
not bugfree (Crunch-Routine could produce Shit without a clear Stack)

********** V1.4 **************
Intuition Interface added, now callable from Workbench, self-detaching,
Req-Lib FileRequester
some Bugs fixed (all code was totally overworked)
```

```
********** V1.5 **************
V1.5a: Demo Version, no Saving possible
V1.5b: Release Version
Both Versions released for the Amiga '91 in Cologne!
Crunch-Routine optimized, now 0,5-2%. better Results!
Menu-Options "Priority" and "LED-Flash" added!
Program sometimes crashed when you quitted it, now fixed
V1.5c:  Auto-SetComment for run-time Data-Decruncher added

********** V1.6 **************
Decruncher added
User-Communication improved
Quit-Bug finally removed

********** V1.7 ** 02-Aug-92 **********
First Release with a free Test-Version and a Registered-Version
New Crunchalgorithms: LZ-Huffman and Quick-LZH
Not available in Address and Link Modes
New Crunchmodes: Data    to crunch Datafiles easily (not via Address-Mode)
                 Switch  which detects Reloc and Data-Files and switches
                         between the Modes (only registered Version)
Speed Improvements: Crunching takes now only half the Time! (only reg.)
Data-Decruncer added
Now Source and Destination Dir can be different
Current Dir of Caller is now the CD of CrM (not df0:)
2.0-style Intuition Interface added
More CLI-Options (including CLI-Help)
New Iconify-Window: shows now ToGo-Counter
Minor Bugfixes (various Actions caused Crashes)
RTDD now removeable, supports new crunchalgorithms

********** V1.8 ** 02-Nov-92 **********
New Decruncher:Library, uses CrM.library
Introducing Library: CrM.library used for executables and data files
Registered Users get Documentation of Library! and a faster one.
Now supports reqtools.library V38! (still uses req.library)
Added Keyboard shortcuts
Added Project Menu
New Mode: Sample-mode, get better ratios with 8-Bit Samples
New Option: Copy Icons (...)
Now supports multiple select possibilities of file requesters
Whole directories may now be crunched
Some changes made to the appearance (Logo, bigger Gads, ...)


              ********** CrM V1.9 ** 08-Oct-93 **********

- Yes! After almost one year another version...

- new GUI: now uses gadtools.library, on Kick1.2/1.3 gadtools13.library is
  used!

- removed possibility to open a window on the wb

- removed  support  of req.library, as there is no new version supporting
  the 2.0-Look
```

```
- new Decruncher: LZH-Algo now also available in Address and Link modes

- added History for Textoutput (only Kick2.0+) (default: 100 Lines)

- added  Percentage Display while crunching, in the main window it's also
  graphical, in the iconify window there is only text.  Ffirst to suggest
  this was Michael

- completely  rewrote  support  for  filerequester, I hope it won't crash
  anymore.   With  multiselect you can now also select dirs, unfortunally
  this doesn't work with mfr (at least it doesn't with my version).

- when  packing entire directories or using the multiselect feature of the
  filerequester  a complete filelist is created before crunching any file.
  Now  Source  and  destination  dirs  may be the same without the risk of
  confusing dos.

- memory  display now uses the correct colors on Kick3.0, thanx to Uli who
  detected this bug and helped me to remove it.

- added menu  : Project / Save Prefs
- added menu  : Options / Encrypt Data
- added menu  : Options / Auto Makedir
- added menu  : Options / Include Subdirs
- added menu  : Options / Text Buffer
- removed menu: Options / Req.library, as CrM no longer supports this lib

- "Copy Icons" still not included, sorry! (menu removed)

- improved user communication.  The display field displays now all actions
  the user has done and looks like a real logfile.

- Reloc/Normal  Decruncher/Kill  Decruncher=Yes  would  let the decruncher
  crash always.  fixed.

- CrM uses and needs now CrM.library V4+

- removed some enforcer hits.


CrM.library v4:
----------------
- added  crunching  algorithms  and  possibility  to  encrypt  data with a
  password

- when  you  enter a password to decrunch some data this pw is stored in a
  list so you have to enter this pw only once per reset.

- speeded up decrunch routines. Optimisation ideas by Fabien Letouzey!

- new functions: cmProcessPW(), cmCryptData(), cmProcessCrunchStruct() and
  cmCrunchData(). See Autodocs for details.


xpkCRM2.library:
-----------------
```

```
- xpk sublibrary using CrM.library v4+

- uses LZ-Huffman algorithm

- first release version


CrMPW v1.0:
------------
- small  tool  to add/remove pws to/from the library internal list without
  using the reqtools.library (affects only decryption)

- provided as assembler example source

- first release version


CrMData v1.0:
--------------
- small command line version of CrM to crunch/decrunch data files.

- supports all algorithms and modes: LZH, Normal, Sample, Password

- provided as assembler example source

- first release version


RTDD v1.9:
-----------
- now  additionally  patches  ExAll(), OpenFromLock() and ExamineFH() on a
  system with Kick2.0+

- now uses and needs CrM.library (no special version required)

- supports  encrypted data:  it automatically asks for the pw and decrypts
  the data.

- new command line parameters: -i: Install Only
                               -r: Remove Only (useful for scripts)


              ********** CrM V1.91 ** 11-Feb-94 **********

- A Null message at CrM’s messageport caused some Enforcerhits. They
  occured when you tried to scroll the output field down. Fixed.

- When  you  wanted  to  quit  from the Iconify window  and you denied the
  requester, CrM would crash.  Fixed.

- CrM caused some Enforcerhits when run from workbench.  Fixed.

- CrM crashed very often when you returned from the Iconify window. Fixed.

- This document is now also available in the amigaguide format.

- Overworked menus: multiple selection (right and left mousebutton) is now
```

possible, Quit shortcut now really only works in combination with SHIFT and the Project menu was reorganized.

- CrM crashed when trying to open a filerequester and reqtools being disabled. Fixed.

- Kicked out "Quick-LZH" mode (actually it was already kicked in V1.8, but it was still in the cycle gadget)

- Changed Task Names to "CrM V1.91 GUI Task.1" and "CrM V1.91 Action Task.1"

- Load/Save/Crunch menus are now also ghosted like the corresponding gadgets.

CrM.library V4.2:

- The BSS segment in combined Data/BSS or Code/BSS hunks was not cleared properly. Thus some programs (e.g. Term) crashed. Fixed.

- Added Oberon interface for this library. Thanx to Alexander Ehlert!

- A mungwall hit frequently occured when crunching in LZH mode, one byte outside the allocated buffer was used. Fixed.

RTDD V2.0:

- RTDD is now a commodity and works only with KS2.0+.

- You can now deactivate RTDD without removing the patches (sometimes you couldn't remove them because another program had also patched some DOS functions).

CrMData V1.01:

- CrMData crashed when the gain was negative. Fixed.

- new option -l: CrMData now defaultly skips files which become longer when being crunched. With "-l" these files may be saved.

xpkCRMS.library V1.1:

- xpk sublibrary using CrM.library v4+

- uses LZ-Huffman algorithm plus Sample Mode (delta encryption)

- first release version

. .   - - -- --- --=- -=-=- -====- -=- -====- -=-=- -=-- --- -- - - . .

If you find any shit or nonsense above this line please tell me. This may occur as I don't read the complete manual each time a new version is released, and as I am really too lazy to care about that...

```
Finito - May the Force be with you!!!

                                      //
                          :-)        //    (-:
                                 \ //
                                  \X/
```