

cfx

COLLABORATORS

	<i>TITLE :</i> cfx		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY		December 8, 2024	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	cfx	1
1.1	CFX DOCUMENTATION	1
1.2	legal	1
1.3	disclaimer	2
1.4	preview	3
1.5	usages	3
1.6	switches	4
1.7	options	5
1.8	bits	5
1.9	output	6
1.10	correspondence	8
1.11	registration	9
1.12	credits	10
1.13	users	11
1.14	rego_form	12
1.15	nico	14
1.16	switch_a	14
1.17	switch_b	14
1.18	switch_c	14
1.19	switch_d	14
1.20	switch_i	15
1.21	switch_k	15
1.22	switch_o	15
1.23	switch_p	15
1.24	switch_r	16
1.25	switch_t	16
1.26	switch_u	16
1.27	switch_v	16
1.28	switch_x	17
1.29	option_a	17

1.30 option_b	18
1.31 option_c	19
1.32 option_d	20
1.33 option_e	20
1.34 option_h	20
1.35 option_i	21
1.36 option_l	22
1.37 option_n	22
1.38 option_r	23
1.39 option_res	23
1.40 option_s	23
1.41 option_u	25
1.42 option_x	26
1.43 option_z	27
1.44 option_dpi	28
1.45 option_gwa	28
1.46 option_gwt	28
1.47 freeware	29

Chapter 1

cfx

1.1 CFX DOCUMENTATION

Everything contained in this release of CFX PRO 5 for the Amiga is USE-AT-OWN-RISK. We, the authors, make no guarantee as to the fitness for use of this software. We accept no responsibility for possible damages incurred before, during or after execution of this software.

LEGAL	Why you are muck for illegally selling CFX!
DISCLAIMER	Why you are at fault when it goes boom!
Preview	Why use CFX?
Usages	Just how do I use CFX?
Switches/Filetypes	What type of files can CFX find?
Options	What other cool things can CFX do?
Bits'N'Pieces	Little bits of hints'n'stuff!
CFX Output	What does CFX's output mean?
Correspondence	How do I contact the authors?
Registration	How much for the registered version?
Credits	Who to thank and why!
Registered Users	The people closest to God!
Registration Form	How do I register?

```
*****
*
*          CFX PRO 5 - Crunched File Xaminer          *
*
*          by BOB RYE and MARCUS MROCZKOWSKI          *
*
*          1st Public Release: Fri 25/9/1992          *
*
*****
```

1.2 legal

DID YOU PAY MONEY FOR THIS SOFTWARE?

CFX is FREeware

This software is FREeware. This software is free. If you have paid money for this software, then you have been ripped off. I have written this program for you, for free. If you paid for "support" from where you got this file, then you are also being ripped off. There are only two people who can fully help you with the operation of this program: Marcus Mroczkowski (co-author), and myself (Bob Rye, the guy who wrote this program.)

You cannot obtain support from some unqualified desk-doofus hiding behind the façade of "a support specialist". All that these people are doing is making money from you, the innocent end-user, and ripping me off by immorally making money from *MY* programs.

Please take a stand against these companies. Please don't buy *FREE* software from these companies. If you would like copies of any (or all) of MY software (anything by Bob Rye) you can have them for *FREE*. That's right, for nothing. All I ask for is a blank disk, to copy the programs onto, and the cost of return postage (and your address!) Then if you want "support" for my programs, send Internet email to me or snail-mail me and you'll get real support.

STAND UP AND BE COUNTED.

Say NO! to costly FREE software.

You may copy and transfer copies of this package to whoever you like, provided that this package is transferred completely intact, with all documentation and executables unmodified therein (modification of copyrighted works is called "contributory copyright infringement" and persons found infringing copyright are liable to legal action); however, if you represent a registered (or otherwise) company, and you wish to redistribute this package, you MUST obtain my written permission. The registered version of the CFX executable may not under any circumstances be copied, (pirated) distributed or given away. The copyright to CFX, the documentation, the source-code, and the actual software remains the explicit property of Bob Rye, © 1988-1994 inclusive.

Distributors of this package must cease distributing this package, if and/or when, specifically requested to do so by the author(s). This request will be made in writing.

ANY commercial distribution of this package without the prior written consent of the author (Bob Rye) is expressly prohibited.

1.3 disclaimer

ALTHOUGH OUTSTANDING BUGS IN THE CODE HAVE BEEN ELIMINATED, THERE REMAINS THE POSSIBILITY OF UNFORESEEN PROBLEMS. WE RESERVE THE RIGHT TO REFUTE THE EXISTENCE OF SUCH 'BUGS', BUT IF FOUND, WE WILL ATTEMPT TO FIX SUCH PROBLEM(S). IF, HOWEVER, UNFORESEEN BUGS ARE FOUND TO CAUSE YOU MENTAL AND/OR PHYSICAL ANGUISH, THEN THAT IS AS THEY SAY IN THE CLASSICS, BAD LUCK! WE ACCEPT NO BLAME FOR ANY LOSS OR

INCONVENIENCE FOUND TO ARISE FROM THE (MIS)USAGE OF THIS PROGRAM. WE RESERVE THE RIGHT TO WITHDRAW SUPPORT AND UPGRADES AT ANY TIME. WE PROBABLY WON'T DO THIS, BUT WE HAVE THIS RIGHT.

ALL INSTANCES OF COMPANY AND/OR PRODUCT NAMES ARE (C), (R) AND (TM) RESPECTIVELY, WHERE APPLICABLE. "CFX" AND "CFX PRO" ARE COPYRIGHT BOB RYE, 1988 - 1994 INCLUSIVE.

CFX VERSION 5 has been extensively tested on:

Amigas 500, 1000, 1200, 2000, 3000, 4000 vanilla PAL, 2.0 - 3.00 OS
B2000 GVP 22mhz & 33mhz ALL-IN-ONE accelerator PAL 2.0 - 3.00 OS
A3000 PP&S 68040 accelerator, 2.04 OS

1.4 preview

CFX requires the ReqTools.library and the PowerPacker.library to run.

CFX was written to enable the Amiga user to identify certain characteristics of given files. The file, let's call it 'TEST', is a file that you received from a BBS, but there were no accompanying docs with it. The file is executable, but you have a feeling that it is crunched. PowerPacker doesn't know this file's 'crunch' type, so you can't easily identify the filetype. You could use a debugger/monitor to find what the filetype is, but the most you'll get out of that is to ascertain that the file REALLY IS crunched. Well, from my experiences on the Aust Amiga Fido Echo, I know that there are a lot of you who don't like receiving crunched files. This program attempts to let you know if it is crunched, and most possibly, what it is crunched with. You can then decide what to do with the mysterious file 'TEST'. Keep it or trash it????

Unlike many other so-called 'file examiners' on the Amiga, CFX has a very large brainfile from which it can acknowledge a large percentage of the current Amiga filetypes. It will then report to you what type of cruncher was used to crunch the file, or, if it isn't crunched, what type of file 'TEST' really is. The information that CFX gives isn't really self-explanatory, so let's describe what the various displays mean.

1.5 usages

Upon invocation with the standard "help" switch:

```
1> CFX ? {return}
```

CFX will show this information:

FA=FindAddress/S,FB=FindBinary/S,FC=FindCrunched/S,FD=FindDocument/S,
FI=FindIFF/S,FK=FindKnown/S,FO=FindOverlay/S,FP=FindPicture/S,
FR=FindRelocator/S,FT=FindTransmission/S,FU=FindUnknown/S,FV=FindVirus/S,
FX=FindExecutable/S,A=About/S,B=BeQuiet/S,C=Crunchfactor/S,D=Deep/S,
DPI=DotsPerInch/K/N,F=FileComment/S,GWT=GetWordText/k,GWA=GetWordAll/k,

H=HunkDisAsm/S, HL=HunksLess/K/N, HE=HunksEquals/K/N, HG=HunksGreater/K/N,
 IF=IntegFail/S, IP=IntegPass/S, LF=LengthFFS/S, LS=LengthSFS/S,
 NV=NoVirusReq/S, REQ=fileREQuester/S, RES=RESource, S=Summary/S,
 SL=SummaryLoud/S, UNC=UNCrunch/S, X=eXplosioninfo/S, Files/M:

As you can see, CFX accepts quite a few options, along with filenames and/or directory/device names. CFX also examines the current directory via:

```
1> CFX {return}
```

First, let us explain the "find" options. When you use a particular option, you are really asking CFX to find a specific filetype. Therefore, when using a CFX "find" option, you must prepend the action specifier with an immediate "f". Here are some examples:

```
1> CFX fa dh1: {return}
```

will examine the root-level of DH1: for any files that are of the address-crunched type. The action specifier in this case is the "a" after the "f". The "a" of course stands for "address". No other filetypes will be displayed other than address-crunched files. By using this option, you are effectively filtering out every other filetype other than address-crunched. (See ADDRESS-CRUNCHED in the FILETYPES chapter)

```
1> CFX ft MAIL: {return}
```

will examine the root-level of your MAIL: directory for any filetypes that fall into CFX's "transfer archive" category, hence the "t" after the "f". (See TRANSFER ARCHIVE in the FILETYPES chapter)

All of the specific filetypes are explained in full in the FILETYPES chapter.

1.6 switches

The following options must be immediately appended to the "f" find switch, thus: CFX fr dh0: {return} (find relocater-crunched)

- a address-crunched
 - b binary (non-executable) files
 - c crunched (address + relocater + transfer) files
 - d document/text files
 - i IFF files
 - k known files
 - o overlaid files
 - p ILBM picture files
 - r relocater-crunched
 - t transmission archives
 - u unknown files
 - v viruses
 - x executable files
-

1.7 options

CFX's also has "support" options, which should make your prospective examinations easier. Let's dissect the options singularly:

Options:

a=about	About CFX
b=bequiet	Be quiet
c=crunchfactor	Crunch factor calculation
d=deep	deep directory scan
dpi=dotsperinch	special picture measurement
f=filecomment	automatic file-comment
gwa=getwordall	finds specified word in ALL files
gwt=getwordtext	finds specified word in TEXT files
h=hunkdisasm	extended hunk disassembly
hl=hunksless	files with < x hunks
hl=hunksequals	files with x hunks
hl=hunksgreater	files with > x hunks
if=integfail	report failed integrity check of files
ip=integpass	report passed integrity check of files
lf=lengthffs	real file length [FFS]
ls=lengthsfs	real file length [SFS]
nv=novirusreq	no requester-warnings for found viruses
req=filerequester	file-requester usage
res=resource	search files for used libraries, devices and fonts
s=summary	summary
sl=summaryloud	loud summary
unc=uncrunch	deep uncrunch/virus check
x=explosioninfo	file eXplosion information

1.8 bits

CFX requires the Nico François ReqTools.library and PowerPacker.library in your LIBS: dir.

The ReqTools.library file-requester accepts extended select (Brett's idea), so use it!

Options and switches may be either case (upper or lower).

Options and switches may be in any order (including pathnames/files).

Filenames/directorynames may be stacked, for example:

```
1> CFX deep dh0: dh1: dh2: {return}
```

will recursively enter and examine all files in ALL levels of drive dh0:, then when finished, repeat the same for drives dh1: and dh2:;

or

```
1> CFX deep c:dir tools:CED dh2:MAIL/trapdoor {return}
```

will examine the files c:dir, tools:CED and dh2:MAIL/trapdoor and then recursively enter and examine all files in drive dh4:

What will happen if you type:

```
1> CFX fr c:dir {return}
```

and the file c:dir ISN'T crunched with a relocater cruncher?? Check it and see if you're right. This same thing will happen if you specify a find option along with asking for the requester. Only specified ('find') filetypes will be displayed if you ask for a specific find-mask.

The AmigaDOS protect bits for the CLI can make CFX's life a misery. If a file hasn't got it's 'R' (Read) bit set, then CFX will report a "READ PROTECTION ERROR" and fail to examine the file. This is not CFX's fault per se, it's the AmigaDOS function "Read()" which won't allow us the privilege, which is just as well, 'cos that's its job! Never mind, just check the protection bits for the file if CFX reports a "READ PROTECTION ERROR" and change the bits via "protect" if you'd like to examine this file. CFX did have the ability to change the bits automatically, but this was not quite right, since you might have unset the "R" bit for a very good reason: file protection. CFX does NOT behave automatically under any circumstances, and does not take matters into it's own hands.

Before launching its activities, CFX checks your DOSBase vectors for intrusion by the Xeno file-virus. This virus is very dangerous if running in a system that CFX is about to utilise. If Xeno is running when you CFX a directory (or if you launch a 'dir' command, or 'list' or ANYTHING that examines directory trees) there is a good chance that some or all of the examined files will be infected. This is the reason that CFX checks the system's current DOSBase. If there is a nasty wedged into DOSBase, CFX will refuse to run, and will explain its decision not to do so. Believe me, you don't want anything like CFX running alongside the Xeno virus.

CFX may sometimes give you the wrong filetype description of a file, because of some utter twist of fate. This is rare. This is unfortunate. This is life. There is nothing we can do about this. Unfortunately, some files "look" like other filetypes, sometimes. Just be happy in the knowledge that CFX doesn't get things confused very often.

CFX version 5.000+ were (and will continue to be) developed with:

- SAS C-compiler, versions 5.10 - 6.51
- SAS assembler 6.51
- DevPac GenIm assembler version 3.01
- Cygnus Editor Professional version 2.12

1.9 output

For the moment, if you think of CFX PRO 5 as just a directory "dir" replacement, then you shouldn't have any problems using it. It's a very informative and comprehensive replacement, however! The standard

CFX output line format is:

```
<      filename      > < size > <flg> <hnk> <   file information   >
```

'filename' is a 28-character field, which will automatically expand with a bigger filename.

'size' is the filesize of the file being examined (defaulting to standardfilesystem size.)

'flg' is the important part here. 'flg' is a 6-character field of flags which tells you everything you needed to know about the file. Let's check what each of the flag-elements means:

1 - the first element can be only one of two things:

'X' or '-'

'X' tells you that the file is 'eXecutable'

'-' tells you that the file is NOT executable (ie. is binary)

2 - the second element can only be one of two things:

'O' or '-'

'O' tells you that the file is 'Overlaid'

'-' tells you that the file is NOT overlaid

3 - the third element can only be one of four things:

'A' or 'D' or 'R' or '-'

'A' tells you that the file is of 'Address-crunched' type

'D' tells you that the file is of 'transmission-arc'd' (DATA) type

'R' tells you that the file is of 'Relocator-crunched' type

'-' tells you that the file is NOT of any discernible type

4 - the fourth element can only be one of three things:

'M' or 'V' or '-'

'M' tells you that the file is a 'Mutant' type

'V' tells you that the file is a 'Virus' type

'-' tells you that the file is NOT a known mutant or virus

5 - the fifth element can only be one of four things:

'F' or 'P' or 'L' or '-'

'F' tells you that the file FAILED a full integrity check

'P' tells you that the file PASSED a full integrity check

'L' tells you that the file is a LINKED file, or, in other words, that this file is the product of a hard link. This flag is only rendered visible when CFX's integrity-checking option is NOT on. This flag entry will appear when CFX finds either a hard-linked file, or a pipe-linked file under AmigaDOS 2.0+.

'-' tells you that a file integrity check was NOT performed

6 - Check the "crunchfactor" option for a description of this flag.

'hnk' is the 'number' of hunks field. This takes the form of 'Hxxx' where 'xxx' equals the number of hunks in the file. Only executable files contain hunks, so binary files won't feature this field. CFX PRO 5 should handle files with as many hunks as is possible.

'file information' is another crucial part of CFX's output. This is where you receive a description string of the file in question. This

may be 'executable type' or any of the other filetype descriptions that CFX currently knows. Here is the CFX output resulting from CFX checking itself:

```
CFX                      48104      X----- H2      executable type
```

Obviously, the filename is 'CFX', its filesize is 48104 bytes (SFS), it's an eXecutable file, has 2 hunks, and is unknown (ie. not a virus or a crunched-file.) Now you should experiment with CFX. Check your entire SYSTEM disk for interest's sake via:

```
1> CFX deep SYS: {return} and ENJOY your program!
```

JPEG (Joint Photographic Experts Group) pictures now have extra information printed beside their output. Normally a JPEG picture would have an output line like this:

```
adpro1.jpg                5127      ----- --      JPEG-JFIF
```

but now CFX can tell you extra information about JPEG pictures. The additional data looks like this:

```
adpro1.jpg                5127      ----- --      JPEG-JFIF (290x348) H8
```

To start with, CFX now can tell you the width and height of your JPEG files. The data after the size is a description of the file's compression method, and whether or not the file is Grayscale (8-bit) or Colour (24-bit).

The 'H' in the above output means that this file was crunched with the baseline or Huffman compression. Alternatively, the letter could be an 'A' which would mean arithmetic compression.

The '8' means that this file is a Grayscale picture. Alternatively the data could be a '24' which stands for Colour.

1.10 correspondence

Snail Mail:

```
BOB RYE
11 BEAVER STREET
ST. ALBANS
VICTORIA
AUSTRALIA 3021
```

Don't forget to include your Internet email address, and personal postal address.

Bob's electronic mail addresses:

```
bob@daemon.apana.org.au
```

1.11 registration

REGISTRATION = \$25 AUS!

If you're using a non-registered version of CFX PRO 5, and you like it (let's face it, what's there not to like??) then please register it. Hundreds of hours, very long hours, have been put into this program. AND it's the best of it's kind, worldwide. Support us and we'll support you. This is the FREEWARE announcement straight from CFX's "about" option. Please read it:

/*-----*/

```
CFX Crunched File Examiner Pro
© Bob Rye & Marcus Mroczkowski   Public
Copyright (c) 1988->1992 incl.    68000
VER 5.226 17:13:59 Jul 14 1992    Version
```

CFX is a file examiner with a difference. It doesn't claim to be anything but a file examiner. Of course, something written today will be out of date tomorrow, but we are committed to making, and keeping, CFX the best file examiner that your precious registration money can buy.

This version of CFX is FREEWARE, which means that if you use CFX, you don't have to provide a small contribution to the authors, but if you do contribute a small amount, you will receive a fully functional version of CFX, with built-in virus-check-uncrunch. We suggest a nominal fee of \$25 AUS (See DOCS), which will not only endear you to us forever, but will also allow you free updates and upgrades whenever they are available. Be daring! Register this FREEWARE product today!

Registration contact:

```
(snail) BOB RYE                      (net) bob@daemon.apana.org.au
        11 BEAVER STREET              PGP2.6ui public key available!
        ST. ALBANS, VICTORIA
        AUSTRALIA, 3021
```

/*-----*/

What do you get for registering?? Here is the short list:

Personal compiled copies, compiled to your specifications (ie. fully optimised, 68000, 68010, 68020, 68030, 68040, all supporting maths-coprocessors.

A clear conscience.

Due to my unemployment, and the fact that I'm flat stoney broke, I cannot cover the costs of postage for updates and upgrades. Your immediate FREEWARE registration contribution does cover the primary media and postage cost, but, unfortunately, I cannot continue to supply a printed manual. If you would like a printed manual, please enclose an additional \$10 AUS. For Air-Mail postage, please enclose an additional \$10 AUS. All registered users may send their original disk, a stamped (PRE-PAID) self-addressed disk-mailer (please quote your registration number) to me at any time and they will receive the

latest version update/upgrade for no cost. To register, fully complete and sign a printout of the registration form at the end of this document, and get it to me, pronto!

1.12 credits

CREDITS & ACKNOWLEDGMENTS

- | | |
|--------------------|---|
| Kim Benson | - for putting up with my very late nights and for feeding me and keeping me warm. There you go, Poss, you've now been read by at least 5 users worldwide! You're famous! |
| Marcus Mroczkowski | - for technical expertise, keeping my Amiga alive, ideas, manual filetype examination, ideas, beer, dim-sims, exhaustive virus-examination and infection, beta-testing, ideas and company. Want a sick computer fixed?? See Marcus. |
| Brett O'Callaghan | - one of THE beta-testers who also contributed positively to the development process, for brilliant ideas, for being intelligent, and for telling me about Larry Niven. (Anyone know where I can get a Nessus doll??) |
| Stephen Walsh | - another beta-tester and sysop of a great BBS and who made constant suggestions for improvement. |
| Martin Ozolins | - a well-hard beta-tester and reg'd owner who, with the help of Enforcer, found several whoppers... |
| Brendan Pratt | - a beta-tester from Qld, with a big BBS to boot. Thanks for the access to SideCar, the files, and the testing! |
| Guns N' Roses | - another spectacular album (or two.) While I can't really admit that Axl actively beta-tested during the recording of UYI 1 & 2, I have heard a rumour that Slash did have a quick tinker during mixing... |
| Van Halen | - thank God Sammy didn't leave Van Halen. Otherwise I couldn't have cruised to 'Judgement Day' from 'For Unlawful Carnal Knowledge'. Could you imagine that pose David Lee Roth singing THAT?? And a word for Eddie, stick to the guitars, mate... |
| Brian and Richard | - thanks for the viruses, guys. I hope the program lives up to all expectations. Talk to you soon. |
| Darius Ignasiak | - who was my first "SHAREWARE" contributor, and who has continued to support the project. Darius, here is your 5 minutes of fame: Darius is the first person in the WORLD to receive a registered (paid) copy of CFX. I hope it gets there on Friday! I also hope you like this Darius... |
-

- Justin Downey - who was the first to comment on the program, and who has dissappeared. Where are you, Justin??
- Mike West - from the center of Australia. Two questions remain on my lips. Where are you Mike?? Had your baby yet?? *;-)
- SAS/C - for providing an **awesome** compiler. Competition? What competition? The first (and foremost) supportive computer/software company that I've ever seen.
- Commodore Amiga - for OS 2.x. This is great. One thing though, where is my developers kit?? Surely you've seen my application by now?? Actually, ANTI-THANKS to C= Australia for their pathetic slowness and irrevocable uselessness. Two years I've been trying to get developer status with these turkeys, and every time I've called them I am told "Mr. Mason is on holidays" or "Mr. Mason hasn't got around to looking at your presentation yet." What crap. Stick it C=. Thank God YOU didn't really invent such a great machine. I'll get my include files from SAS in the future.
- Nico François - A special thankyou goes to Nico François, whose ReqTools.library makes a program look, feel and behave supremely, and whose PowerPacker.library makes uncrunching things extremely fluffy! Thanks Nico!
- Anthony Horan - Another bug hunter extraordinaire! Just call him Mr. Internet!
- Jean-Guy D. Speton - An Internet "filetype-informant" who supplied several bits of great information, especially ByteKiller 2. Pity more users aren't as informative as Jean-Guy! Thanks dude!

1.13 users

Marcus and myself would like to thank the following people:

1. Stephen Walsh
 2. Darius Ignasiak
 3. Mike West
 4. Brett O'Callaghan
 5. Richard McElvenny
 6. Justin Downey
 7. Brendan Pratt
 8. Amiga Quarantine
 9. Andrew Garth
 10. David Margerison
 11. Wolfgang Schoenfelder
 12. Commodore Australia (Ha!)
 13. Ron Hocking
-

14. Andrew Rogl
15. John Paul Lonie
16. Andrew Leniart
17. Chris McCallum
18. Ian Rodbourn
19. Chris Quonoey
20. Martin Ozolins
21. Kenneth Simpson
22. Geoff Lake
23. Anthony Horan
24. Pat Caldon
25. Rob Nottage
26. Claus Pedersen
27. James Abram
28. Howard Alexander
29. Erik Løvendahl Sørensen
30. Robert Byrne
31. Andrew Clayton
32. Colin Fraser
33. Damon Pillinger
34. Steven Sheedy
35. Parrish Heywood
36. Martin Stannard
37. Jean-Guy D. Speton
38. James Ray
39. Gavan McCormack
40. Terry Wood
41. Neville Duguid
42. Niels Van Willigenburg
43. Olaf 'Olsen' Barthel
44. Warrick Burgess
45. Heikki Kantola
46. Jave Ivanovski
47. Brian Myers
48. Dave Dustin
48. Heinz-Guenter Boettger

Updates are available whenever you feel like one!

1.14 rego_form

MAIL TO:

BOB RYE
11 BEAVER STREET
ST. ALBANS
VICTORIA
AUSTRALIA 3021

CFX REGISTRATION: I (Bob Rye) will not accept monies from any person in payment for registration of CFX if the person's signature is not included on this form. You have been warned.

Please read the registration page and this notice fully:

DISCLAIMER

ALTHOUGH OUTSTANDING BUGS IN THE CODE HAVE BEEN ELIMINATED, THERE REMAINS THE POSSIBILITY OF UNFORESEEN PROBLEMS. WE RESERVE THE RIGHT TO REFUTE THE WORD OF THE USER AS TO THE EXTENT OF SUCH 'BUGS', BUT IF FOUND, WE WILL ATTEMPT TO FIX SUCH PROBLEM(S). IF, HOWEVER, UNFORESEEN BUGS ARE FOUND TO CAUSE YOU MENTAL AND/OR PHYSICAL PAIN, THEN THAT IS AS THEY SAY IN THE CLASSICS, BAD LUCK! WE ACCEPT NO BLAME FOR ANY LOSS OR INCONVENIENCE FOUND TO ARISE FROM THE (MIS)USAGE OF THIS PROGRAM. WE RESERVE THE RIGHT TO WITHDRAW SUPPORT AND UPGRADES AT ANY TIME. WE PROBABLY WON'T DO THIS, BUT WE HAVE THIS RIGHT.

If you would like to continue with registration, please fill in the below. The registration fee is \$25 Australian.

NAME:_____

ADDRESS:_____

ELECTRONIC ADDRESSES: (if any, ie. Internet Email etc)

MACHINE CONFIGURATION:

(ie. 500, 600, 1000, 1200, 2000, 2500, 3000, 4000, accelerator etc)

I hereby attest to my understanding of the above terms by giving my signature:

SIGNATURE:_____

DATED:_____

Depending upon mode of mailing chosen (by you, read the registration

page!) please allow up to eight weeks for delivery of your copy of CFX.

1.15 nico

ReqTools.library and PowerPacker.library are © (copyright) Nico François. Many thanks go to Nico for his great libraries!

1.16 switch_a

This switch will force CFX to only display files that have been output by an address-cruncher. Address-crunchers tend to be the scourge of the computing society, so these files shouldn't really be allowed to live, unless of course, you really need the file. CFX knows most of the Amiga address-cruncher formats. CFX doesn't currently check the uncrunched contents of all of these filetypes, although some are now covered (See "unc - deep Uncrunch" option later). This switch cannot be used in conjunction with any other switch, although some options may be used. See BRAINFILE INFORMATION for more info on address-crunchers.

1.17 switch_b

When this switch is used, CFX will only display information on non-executable files. This filetype covers all non-executable filetypes, including transmission archives, text files, picture files etc. This switch may be used in conjunction with the "known files" and "unknown files" switches, and some options. Obviously, you cannot ask CFX to display only executable binary files, as binary files are NOT executable in the eyes of CFX!

1.18 switch_c

This switch will allow CFX to display all files that it finds that have been crunched by something. This includes Address-crunched files, Relocator-crunched files, and Transmission archives, hence the (a+r+t) after the description. This switch cannot be used in conjunction with any other switches, although it is possible to mix this switch and some options.

1.19 switch_d

This switch will allow the user to find only plain document/text files.

1.20 switch_i

This switch allows CFX to find and display only IFF format files. Practically all IFF files are catered for in this switch. If CFX finds an IFF file that it doesn't actively know or recognise, then it will print "Unknown IFF format". If this happens, please send us a copy (or the first 4000 bytes) of the file(s) in question and we'll add it to our brainfile. This switch cannot be used in conjunction with other switches, but may be used with some options.

1.21 switch_k

This switch is for use in conjunction with one of these other switches: <NONE> <BINARY> <OVERLAYED> <EXECUTABLE>. When used without another switch (ie. CFX fk {return} or CFX fk dh3: {return}) CFX will only display files that it knows. Considering that the list of filetypes that CFX knows is rather extensive, you should find a lot of files! When used with the binary switch (ie. CFX fk fb {return}) CFX will only display files of non-executable type that it actively knows. This could cover filetypes of known IFF's, or certain configuration filetypes etc. In conjunction with the overlayed switch, this switch will find known overlayed files. This could include the "DImp executable disk archive" or the "PowerPacker 3 overlayed file crunched file". Finally, you may use this switch with the executable switch. This will ensure that CFX only displays executable files of known type. This could include all crunched executable files that CFX knows.

1.22 switch_o

This is a very handy switch that allows the user to find overlayed filetypes. Since it may be difficult for the non-technically minded user to find the overlay id long-word in an executable file-header, this option allows you to find which files are overlayed. Overlayed files use a special technique for loading into the Amiga's memory. Certain software producers use this technique on their larger files so as to not rob the Amiga of vital memory during loads (ie. instead of loading a 340k file directly into the Amiga's memory, an overlayed file may only load in what parts of the file that they need at any one time. Therefore this file may only load 50k at any one time, thereby freeing about 300k of memory for other uses.) Obviously, this switch finds executable files (only they can be overlayed!) Therefore, no other switches work with this one, other than the "known" and "unknown" switches, and some of the options.

1.23 switch_p

This is almost a subset switch of the IFF switch. This allows CFX to find and display ONLY IFF picture files, and not *ALL* of the entire

IFF gamut. Also, all IFF (ILBM and ANIM) picture files found will be displayed with normal and additional information (if possible.) The picture's size and number of bitplanes will also be displayed, making this switch an excellent tool in itself. This switch is mutually exclusive from other switches, but may be used with some options.

1.24 switch_r

CFX will display files of this type if this switch is used. Whilst similar to address-crunched files in some ways, relocater-crunched files tend to be more useful, and more kind to the machine. Some of these relocater-crunched files actually use more intelligent scatter-loading methods than AmigaDOS itself (wouldn't be hard, though *;-) This switch can only be used by itself, with a couple of the options thrown in for good measure. After all, you don't want to find all of the files that are address-crunched relocater IFF's, do you??

1.25 switch_t

This is another handy little switch for finding (and examining, to an extent) files that are considered transmission archives by CFX. Since most of you use these archiving methods, I won't go into the "Complete works of Lempel and Ziv", or explain the inner workings of "adaptive Huffman encoding". Instead, let it suffice to say, that CFX knows lots of 'em. If you have any of these cute archive types that CFX doesn't know, then please let us know! This switch is mutually exclusive with other switches, but some options may be used with it.

1.26 switch_u

See "known files" and reverse what has been said.

1.27 switch_v

Unluckily, the Amiga is following the rest of the computer/PC cliques (or should that be cliches??) that have hordes of rabid spotty juveniles writing heroic viruses (I think I must attribute that line to Andrew Clayton!) These authors are of course witless beyond belief. Luckily, we have been allowed to obtain several file-virus filetypes from Amiga Quarantine, in New South Wales. The guys at A.Q. (Richard and Brian Logan) were gracious enough to allow us access to some of the common (and some uncommon) nasties that might visit you, if you're not careful! With this switch, you can force CFX to ONLY check each file found for virus infection. This will ensure that CFX ONLY displays information about files that are of a virus filetype. You don't, however, have to manually use this switch for CFX to virus-check your files. CFX automatically checks each executable file

for infection EVERY time it is run. Since CFX is a hybrid of assembly and C, the asm virus checking routines are lightning quick, hence the automatic checking. When/if CFX finds a virus-infected file, it will print its usual information, AND display a requester informing you of your infection (ooh, sounds nasty.) CFX doesn't remove or disinfect infected files. It is only there to find such infections. So far, CFX has performed beautifully, and has even outperformed some of the latter-day anti-virus utilities. Another thing to be aware of, is CFX's built-in anti-virus protection. CFX has two intelligent routines that it performs every time CFX is invoked. Firstly, CFX checks it's filename (argv[0]) and if this isn't "CFX" then it will bomb out, with a warning. This is for protection against those "replacement" viruses which rename your original file to something like "DEVS: ". Secondly, CFX performs a self-anti-virus test upon each invocation. If CFX finds that itself has been infected, CFX will not run further, as doing so may infect each and every file that CFX checks! Not good. Therefore, CFX needs to be able to find itself upon every invocation. This means one of two things: you MUST place your copy of CFX in a system-pathed directory (ie. C:) or you must call CFX with it's full pathname (ie. DH1:TOOLS/CFX.) The latter is cumbersome, so use the former method. If CFX cannot find itself, through pathnames, or through tracing its own argument vectors, then it will run without checking itself. This is your problem. Also, from version 5.267, CFX will notify the user audibly that a virus has been found in any way. Even if the user has selected 'nv' for No Virus Warning Requesters, CFX will still beep you when a virus is found.

```
*****
*** In conclusion, only use fv if you are specifically looking ONLY ***
*** for infected files, as CFX will always check ALL executables for ***
***                               infection anyway.                               ***
*****
```

1.28 switch_x

Will force CFX to only display executable files that are found during its scan. This switch can be used with the "known" and "unknown" switches. You can't use this switch with many other switches (other than find unknown and known), but some options are available with this one. This switch is useful for finding files which you can run on an unknown disk (AND CFX performs a virus check on each executable that it finds!)

1.29 option_a

This option shows one of two things. If you have paid a registraton contribution to the authors, you will receive a personal version of CFX. This version will have your name and other information in this screen:

```
CFX Crunched File Examiner Pro
© Bob Rye & Marcus Mroczkowski   Registered
```

Copyright (c) 1988->1994 incl. 68030
VER 5.229 16:47:28 Aug 10 1992 Version

Compiler: SAS/C version 6.50

This CFX is registered to: Bob Rye

Registration contact:

(snail) BOB RYE	(net) bob@daemon.apana.org.au
11 BEAVER STREET	PGP2.6ui public key available!
ST. ALBANS, VICTORIA	
AUSTRALIA, 3021	

Otherwise, if you are still using the public release version, this screen will explain the FREEWARE concept, and give you information about how to contact the authors for when you decide to register CFX.

CFX Crunched File Examiner Pro
© Bob Rye & Marcus Mroczkowski Public
Copyright (c) 1988->1994 incl. 68000
VER 5.226 17:13:59 Jul 14 1992 Version

CFX is a file examiner with a difference. It doesn't claim to be anything but a file examiner. Of course, something written today will be out of date tomorrow, but we are committed to making, and keeping, CFX the best file examiner that your precious registration money can buy.

This version of CFX is FREEWARE, which means that if you use CFX, you don't have to provide a small contribution to the authors, but if you do contribute a small amount, you will receive a fully functional version of CFX, with built-in virus-check-uncrunch. We suggest a nominal fee of \$25 AUS (See DOCS), which will not only endear you to us forever, but will also allow you free updates and upgrades whenever they are available. Be daring! Register this FREEWARE product today!

Registration contact:

(snail) BOB RYE	(net) bob@daemon.apana.org.au
11 BEAVER STREET	PGP2.6ui public key available!
ST. ALBANS, VICTORIA	
AUSTRALIA, 3021	

Surely for this much work, we deserve something, don't we??) This option is best used by itself.

1.30 option_b

This option will allow you to dodge the CFX title/heading which CFX prints at the start of each session. Why this option? If you use Directory Opus (quite fantastic IMO) and you have CFX hooked into a tool gadget, you will soon realise that DOpus uses (or did use, I don't know if it's changed!?) a 256-byte string buffer for selected filenames. If you picked 23 filenames in DOpus, and the length of all these filenames exceeded 256 bytes, then another string buffer would be allocated for you. Therefore, CFX would print its title, and then do its work, and

then get its next string buffer and do it all over again. Seeing the CFX title after every 6 files have been CFX'd gets a bit annoying. This option alleviates this somewhat.

1.31 option_c

This option used to be the checksum calculation option, but after thinking about its redundancy, I changed this to something more useable and important (I think). Using this option will slow CFX down slightly, but its output may help you to decide whether a file is crunched or not. This may be best used on unknown files, which CFX cannot explain, or that are new to your system. CFX will use the new 6th flag in its printout for this option. This option will allow CFX to output a number, scaled to the file's crunched-ness (or thereabouts). Here is an example:

```
brain.doc                88290      -----0 --   ascii text
```

I have just run "1> CFX c brain.doc" and the above was output. You should be able to figure out all of the output except for the "0" at the 6th flag. This "0" means that CFX thinks that this file is NOT crunched. CFX thinks that you could actually decrease this file's size by (re)crunching this file. The only problem I have surveyed about this function is that CFX sometimes miscalculates SMALL crunched files as being a crunched-ness of "0" where they may really be quite crunched already. This ONLY appears to happen on SMALL crunched files. Here is another example:

```
CFX_931007.shr           179546    --D--9 --   IFF-CDAF Compressed data
```

This archive of CFX's sources is also quite easy to understand, but for the 6th flag. The "9" means that CFX doesn't think that this file can be crunched any further using standard (ie. normal) crunch techniques. CFX is RARELY wrong on type-9 files. Here is a table of possible CFX outputs using the "c" option:

- U - Means that CFX couldn't fully examine this file. This could mean a file I/O error, or just an empty file.
 - 0 - Means that CFX thinks this file is NOT crunched at all.
 - 1 -}
 - 2 -}
 - 3 -} Means that CFX considers the file not to be crunched, or crunched } so minimally that further size reduction is certainly possible.
 - 4 -}
 - 5 -}
 - 6 - CFX considers that file may be VERY loosely crunched, if at all.
 - 7 - Again, this file maybe crunched, but again only minimally.
 - 8 - This number usually points to the file being crunched with a file-cruncher type of cruncher, like Imploder, PowerPacker or TetraCrunch. This also may point to a loosely crunched *.lha file. Further size reduction via crunching will be minimal at best.
 - 9 - CFX considers this file to be quite optimally crunched. Further size reduction on this file will probably be impossible. Examples of this filetype are normal *.lha, *.dms, *.shr or *.gif files.
-

You can of course cross-check CFX's findings by attempting the recrunch any of these filetypes. If they crunch well, and CFX says they might, then great. If they don't, then blahh!

1.32 option_d

One of the most basic (and important) options. The "deep" will allow you to recursively enter directory levels. If you were to use:

```
1> CFX deep dh0: {return}
```

CFX would first examine the root level of dh0:, and then each of the sub-directories from the root. The above example would provide an examination report for ALL files and directories on drive dh0:. This option can be used to modify all action switches and options. When CFX outputs its findings, it doesn't sort the directory entries in any way, as this would entail buffering and sorting before output. We believe that reading a directory structure straight from DOS is the best way to get to filenames, hence no sorting/pausing. CFX now has a limit as to how many directory-levels it can descend. As of this writing, the number has been set at 48 levels, although this may change at any time, for bigger or smaller. This has been implemented because of the fact that CFX recurses, and uses more memory each time it enters a new directory-level. This memory comes from the system-stack. If you have a low system-stack when you use "deep", and CFX crashes, then you should set your stack to a higher number. The bigger the stacksize, the better. To allow for the recursive collecting of 48 layers of directories, you need 30000 bytes of stack. You have been warned!

1.33 option_e

Another option for the technically minded. This allows the user to FULLY examine the hunk structure of a specified file(s). Of course, this will only be effective with executable files, so non-executable files won't provide an extended-hunk report. This can be useful for checking whether a file contains annoying debug hunks, symbol hunks, or anything else hunk-oriented. You can use this option with most other options and switches, although extended hunk-information won't be provided for non-executable files.

1.34 option_h

This option is practically a "find" switch camouflaged as an option. If you would like to know which files on your disk contain a certain number of hunks, this is the option to use. Some programmers release executables containing innumerable hunks. This can be good and bad. If your system contains severely fragmented memory, a program of 97 hunks will probably scatter-load a bit easier (memory permitting) than a 2 hunk file of the same size. Imagine your Amiga's memory as being a big box containing several pigeon-holes. When the memory isn't fragmented,

there might only be 4 pigeon-holes, that are quite large. When fragmented, the pigeon holes are more in number, and a fair bit smaller. A large program of two hunks will need a certain amount of contiguous memory to scatter-load. If fragmented, your machine may not be able to load the file. On the other hand, the fragmented machine may be able to load the same sized program of 97 hunks, as each hunk is small, and easily fitted into the "broken" memory. If you are an avid Imploder 4 user, then you may want to "hunk-merge" such programs with huge numbers of hunks. If you need to know whether your TOOLS: directory contains any files of more than 10 hunks (my personal cut-off level), then CFX the directory with:

```
1> CFX hg=10 TOOLS: {return}
```

CFX will then inform you of the files of more than 10 hunks in your TOOLS: directory. You can also find files containing a specific number of hunks by using the "he" operator, or files containing less than a certain number through the "hl" operator. Hence:

```
1> CFX hl=3 TOOLS: {return}
```

will find files of less than 3 hunks in TOOLS:. Of course, you cannot specify an illegal formula like:

```
1> CFX hl=1 TOOLS: {return} or
```

```
1> CFX hl=-10 TOOLS: {return}
```

as executables cannot contain less than 1 hunk. The numbers that you give to CFX for this option MUST BE positive decimals. This option can be used with many other options and switches, although useless if used in conjunction with "fb" for instance. If you don't understand this concept, Addison-Wesley print excellent books for the Amiga...

1.35 option_i

Have you ever received a disk of pictures from someone and noted that some of the files have disk-errors right in the middle of them? Have you also noted that the pictures with the errors in their guts are the ones that you REALLY want? Just as Mostra is about to show you the incredibly viable HAM interlaced version of "KELLY_BUNDY_GOES_NAKED_SEX_FEST.pic" you hear that sickening <grrr><grrr> noise from DF0:. Oh, I'm all a-swoon from the thought of it! This can provide you with quite a lot of teeth-gnashing. The "ip/if" option checks all specified files on a disk for these horrid read/write errors. If CFX finds a file that has compromised integrity, CFX will report this with a flag in its output. (See "CFX OUTPUT" later...) You then know that this file is knackered. Brett O'Callaghan then asked for more from this option, so he got it. Brett said that he'd like CFX to be able to report just WHAT files passed or failed an integrity check, while skipping other irrelevant information. Well, this function is now implemented, as you can now specify either:

```
1) a straight integrity check of all files, with reporting of all file's statuses, or;
```

- 2) an integrity check of all files, but only receiving reports about which files FAILED the check, or;
- 3) an integrity check of all files, but only receiving reports about which files PASSED the check.

To gain a complete integrity check of all files (and receive full reporting), use the 'ip/if' option. For a full check, but only receiving reports for FAILED or PASSED files, use 'if' or 'ip' respectively. This option is subject to all other dynamic switches/options, as in using the 's' option with the 'if/ip' option, you won't receive any information (unless you use the BOC 'sl' option) about individual files, but only a summary report. Then check the number of entries in the 'Compromised files' area for how many bad files there are. This option can be used with most other options and switches.

NOTE: Attempting to check an empty file's integrity is not possible (why bother?) Thanks to Martin Ozolins who made some chilling discoveries about CFX...

1.36 option_l

"Bob, I have a cunning plan." This is what Brett O'Callaghan once said in response to the completely blasphemous way that AmigaDOS handles filesizes. I won't explain the way that DOS handles the filesize information, but suffice it to say, it's stuffed (just like Baldrick's plans.) Brett came up with a plan so cunning, you could cut your lunch with it. This involved asking DOS how big a file is, and then rolling it in a cute little algorithm of mine. Now, using this option, CFX will list filesizes CORRECTLY. That is, instead of being told that your disk containing one hundred 3-byte files contains 300 bytes of information, CFX will tell you that you really have a lot less than 835.7 k free. Don't forget, that each of these 3-byte files is really exactly 1-kilobyte long (one parent block = 512-bytes, one data block = 512-bytes). This option defaults to standardfilesystem sizes. You can use fastfilesystem sizes via:

```
1> CFX lf ... {return}
```

I don't know how useful this option is, as it would be handier in a directory-handling program (like that excellent Australian one with the author who doesn't reply to netmail.) This option is also safe to use with most other options and switches, as its actions are practically passive.

1.37 option_n

This is for those of you who like to redirect your outputs. When CFX finds a file-virus on a disk, it reports this via a big requester. Since this requester will sit there on your CLI for an eon or two, you can turn off the requesters. Now you can check a disk for viruses,

whilst redirecting CFX's output, without having to worry about clicking requesters every 7 seconds. Another excellent idea by CFX's co-author, Marcus Mroczkowski. This option can be used in conjunction with other options and switches, although you must be careful with its use, as CFX would (by default) warn you that it has found a virus on your disk. If you turn off the requester-warning, then pay attention to CFX's file-information area, as CFX will ALWAYS print to the standard output that a file-virus has been found.

1.38 option_r

For those people who like to point and click. This version of CFX allows a *MUCH* better representation of a file-requester, using the brilliant Nico François ReqTools.library requester (makes the ARP and REQ requester look very dense.) By using the command-line:

```
1> CFX req {return}
```

CFX will bring up the requester, and ask for your selection(s). You can either change directories or sit where you are; point and hit "OK"; double-click a filename; or extended-select a group of files (thankyou Brett!) using the shift-key for the extended-selection modifier (ala WorkBench), select ALL of the files in the current directory by hitting the "ALL" gadget, select files via a wildcard arrangement, using the "MATCH" gadget; or just hit "CANCEL" and do nothing.

1.39 option_res

I have finally been convinced to include a resource checker in CFX. CFX will check for any libraries, devices and fonts/otags that an executable file uses via:

```
1> CFX res path/file {return}
```

By using the "deep" option, CFX will recurse through directories as per usual, but it will report on found strings. This can be useful as a "pre-SnoopDos" check of files before running them, to check what libraries etc. that you may need. Because of the incredibly open programability of the Amiga, these library, device and font names may be practically anything. CFX will do its best to figure out just what IS a proper name, and what isn't. Also, CFX will not doubly report the same string twice, and will sort the output for your ease of reading.

1.40 option_s

When examining a whole directory tree, the user may not want or need CFX to display everything about the files found. You may only want a summary of valid points about the dir-tree. This information can be very interesting, and is generally quicker than a full-display of CFX's

work. Here is a summary of my DH1: partition that CFX gave me after the command line:

```
1> CFX deep summary dh1: {return}
```

```
CFX Crunched File Examiner Pro
© Bob Rye & Marcus Mroczkowski
Copyright (c) 1988, 89, 90, 91
VER 5.08p 14:53:02 Oct 20 1991
```

```
CFX Summary of "dh1:" (HD_1):
```

```
-> Relocator-crunched.....: 74          Size : 1225988
-> Address-crunched.....: 1          Size : 6164
-> Other known executables...: 0          Size : 0
?? Unknown executables.....: 115       Size : 4104876
```

```
-----
Executable files checked.....: 190       Size : 5337028
```

```
-> Crunched data files.....: 1          Size : 54708
-> Other known data files...: 4          Size : 68606
?? Unknown data files.....: 0          Size : 0
```

```
-----
Data files checked.....: 5          Size : 123314
```

```
Acceptable files.....: 195          Size : 5460342
Compromised files.....: 0          Size : 0
Directories found.....: 7
FILE/LINK VIRUSES found.....: 0
```

```
Total % of known files.....: 41.03%
```

The only other thing I should explain is the [LOUD] sub-option. With the above example, (and as is usual) CFX used the default QUIET option. You can force CFX to print it's usual information display whilst doing the summary scan. The LOUD option must be specified immediately after the "s". (Thanks again to Brett O'Callaghan for this little titbit!) This option is particularly useful when used in conjunction with some of the "find" switches. Of course, the percentage known total at the bottom of the summary output can appear silly at times. For example:

```
1> CFX fr d s dh0: {return}
```

will search (deeply) through drive dh0: for all relocator-crunched files, and then display a summary screen on the found files. Of course, the % known field will be 100% if any files were found. If you can't figure out why this is so, go back and use Whatis.

NOTE: CFX now allows stacked device/dir/file names to be passed to the summary function. This means that the following command line is perfectly valid:

```
1> CFX fr d s dh0: dh1: dh2: dh3:{return}
```

as long as you don't include too many name directives. CFX now ceases counting at 30 files/devs and will only examine the first 30 given.

One thing you should also be made aware of is the fact that CFX's switches and options are NOT case-sensitive. So, either FA, Fa, fA, or fa will gain the same action. Another caveat is the argument positioning. CFX doesn't need arguments in any particular order. For example:

```
1> CFX s d fr dh0: {return}  is identical to:
1> CFX d s fr dh0: {return}
```

If CFX gives you this warning:

```
CFX Crunched File Examiner Pro
© Bob Rye & Marcus Mroczkowski
Copyright (c) 1988, 89, 90, 91
VER 5.08t 14:18:14 Oct 23 1991
```

Argument conflict: illegal combination of switches.

then you know that you have offended CFX's logic. If you have a mixture of switches that CFX refuses to accept, that YOU think should be OK, then please email/netmail me with your ideas. I'll certainly have a look at them.

1.41 option_u

This is a new option which will definitely grow as we add more cruncher types to this function. What this does is quite simple (from the user's viewpoint!) If CFX finds a file that has been crunched by a certain cruncher type that it can uncrunch, and the unc option has been given, CFX will attempt to uncrunch the file, and then scan the uncrunched memory for viruses. Currently this option supports these cruncher types:

```
ByteKiller 1.2-1.3
Tuff 1.0
Tetragon "Tetrapack" 2.1 - 2.2 +PRO
DefJam 3.2
PowerPacker data
PowerPacker 2.x OTHER THAN MASTER modes
PowerPacker 3.x OTHER THAN MASTER, overlayed and encrypted modes
PowerPacker 4.x OTHER THAN overlayed, loadseg and encrypted modes
All Imploders, 1.0-4.0 OTHER THAN S-Lib & L-Lib crunched, overlayed
crunched, and some mutants.
```

Technically speaking, ByteKiller, Tuff, Tetragon(s) and DefJam(s) usually stomp memory where they uncrunch, but CFX doesn't allow this to happen. CFX allocates a safe memory area (memory permitting) and allows these crunched files to uncrunch safely. The uncrunched programs in this mode DO NOT launch, so this function is safe. If a virus is found hidden in a crunched file, CFX will give the appropriate warnings. Of course this option can take some time to fully execute if you have 300 (supported) crunched files in a directory to check. Usage example:

```
1> CFX unc d DH1: {return}
```

will examine ALL directory levels of drive DH1:, and if CFX finds any of the supported crunchers, it will attempt to uncrunch them, check them, and report it's findings. Sometimes CFX may report that a file was too mutated for uncrunch-examination: this is just tough luck. CFX will then skip this file. One file comes to mind, and that is MandelMountains 2.1 (mutated like the be-jingos.) To uncrunch this file, run two copies of Zap, load in MM2.1 into one Zap, and another Imploder Normal file into the other Zap. Now, after the longwords \$000003E9, \$00000014 in MM2.1, copy Normal Imploder code into the MM2.1 Zap window, until you hit the first \$000003F2. Now use PowerPacker 3.0+ to uncrunch this file. Experiment...

On registered versions of CFX Pro, some MUTANT Imploder filetypes were so mutated that CFX had NO real chance of uncrunching them for the file-virus check. This has still not changed, and any attempt to use the "UNC" option on mutated files may result in a severe system crash. CFX has, however, been changed so that it now attempts to determine the extent of mutation, and if found to be severe, CFX will not uncrunch the file. This does not preclude the possibility that the crunched file is knackered, and CFX may still crash the machine when attempting such examinations, even though CFX now has tighter control on the situation. This now also applies to the PowerPacker MUTATIONS as well. CFX will mostly uncrunch and virus-check these files, but if the PowerPacker file is comprehensively mutated, CFX may unfortunately crash your system. Please note that this hasn't happened on my machine for quite some time, and I regularly virus-check crunched files of this type...

1.42 option_x

This is a new option which will investigate/disassemble certain files. The file explosion ability can be quite handy, since it peeks into several archive/crunched-file types. An example of this would be:

```
1> CFX x DH1:SRC_BAK/CFX5_92_08_11.lha {return}
```

```
CFX Crunched File Examiner Pro    Registered
© Bob Rye & Marcus Mroczkowski    68030
VER 5.229 16:47:28 Aug 10 1992    Version
```

```
<BASE DIRECTORY> "DH1:SRC_BAK" (BAK)
CFX5_92_08_11.lha                204595  --D-- --    Lharc -lh5- arc
-> F = 87    h0 = 0    h1 = 0    h2 = 0    h3 = 0    h4 = 0    h5 = 87    h6 = 0
-> h7 = 0    hd = 0    z4 = 0    z5 = 0    zs = 0    ?? = 0    T = 494490

    1 files    1 directories    204595 bytes    CFX finished!
```

The key to the above format:

```
F  = number of files found in this archive (whatever the format.)
h0 = number of stored (-lh0-) files in this archive
h1 = number of -lh1- files in this archive
h2 = number of -lh2- files in this archive
h3 = number of -lh3- files in this archive
h4 = number of -lh4- files in this archive
h5 = number of -lh5- files in this archive
```

h6 = number of -lh6- files in this archive
 h7 = number of -lh7- files in this archive
 hd = number of -lhd- files in this archive

z4 = number of -lz4- files in this archive
 z5 = number of -lz5- files in this archive
 zs = number of -lzs- files in this archive

?? = number of unknown-format files in this archive

T = total uncrunched/exploded size of files in this archive

The filetypes that CFX can disassemble are many varied. Obviously, LHA and LZH files (along with quite a few *.LZ* variants) are supported. NOTE: This option was designed with only level-0 headers in mind: I have since re-written this disassembler with several things in mind. One was to reduce the size of the resulting code; two was to speed things up a bit; and three was to incorporate a multi-level LHA/LZH archive disassembler. All three points have been met successfully. Also supported are several disk-crunchers, like DMS. Address-crunchers are also supported, like:

```
1> CFX x CFX:ETEST/tetra21_plain {return}
```

```
CFX Crunched File Examiner Pro
© Bob Rye & Marcus Mroczkowski   Registered
Copyright (c) 1988->1992 incl.    68030
VER 5.229 16:47:28 Aug 10 1992    Version
```

```
tetra21_plain          4948      X-A-- H1      Tetrapack 2.1 norm-decrunch
-> UNCR AREA = $30000-$50000  JUMP ADDR = $30000  EXPL. SIZE = 8822
```

```
1 files  0 directories  4948 bytes  CFX finished!
```

The key to the above:

UNCR AREA [Uncrunch Area] = hex address \$30000 - \$50000. This is the area of your memory that will be bulldozed when this address-cruncher expands.

JUMP ADDR [Jump Address] is where execution of the newly expanded data will commence.

EXPL. SIZE [Exploded Size] is how big this crunched file will expand to inside the Amiga's memory.

This feature allows you to check files more deeply, and may be confusing to some users. If so, don't let it worry you. This option may not even be of any use to you.

1.43 option_z

This is a new option which allows CFX to automatically file-comment each file that CFX checks. Each file checked by CFX will have its comment field changed to signify the filetype that CFX thinks the file is. This

doesn't take any extra disk-space for the file. You can then check the file's filetype simply by using the AmigaDOS "List" command on the newly commented file. Be aware, however, that CFX is now writing the comment information to your disk, so please observe the normal safety precautions in regards to writing to a disk.

1.44 option_dpi

This is a special option which allows you to specify a DPI (Dots Per Inch) measurement. When CFX finds a picture file, it attempts to find the x (width) and y (height) size of the picture. If CFX can find these numbers, and you do specify a DPI number, CFX will divide the x and y sizes of the picture by the DPI number specified. This will give you a set of measurements which you can use in desktop publishing software. For example, if you have a 300-dpi printer, and are using PageStream and you want to print out this picture at your printer's highest resolution, then run CFX dpi=300 on the image file to find out the corresponding inch size of the image. This size is based upon the picture's x and y sizes in pixels. The PageStream command is "Edit Coordinates" (rAmiga-E) to scale the image to the correct size in inches. You can then type in what CFX tells you about this picture into PageStream. Phew. This was Damon Pillinger's idea, so blame him! ;)

1.45 option_gwa

Every GREP I try does something stupid. Most are ok, but most also don't recurse into directories. I don't really need all of the strange wildcarding classes of a full GREP, and would only ever use a word search if available. This option allows CFX to search each and every file it finds for a specified word. The difference between this and the GWT option is that this option searches ALL files, and not only TEXTS. The word's case is unimportant to CFX. For example:

```
CFX gwa=Picasso deep uucp:news {return}
```

will list ONLY files containing the word "picasso" or "Picasso" or "PICASSO" etc. You get the idea! Embedded spaces are NOT allowed, sorry.

1.46 option_gwt

Every GREP I try does something stupid. Most are ok, but most also don't recurse into directories. I don't really need all of the strange wildcarding classes of a full GREP, and would only ever use a word search if available. This option allows CFX to search all TEXT files that it finds for a specified word. The difference between this and the GWA option is that this option searches only TEXT files, and not ALL files. The word's case is unimportant to CFX. For example:

```
CFX gwa=Picasso deep uucp:news {return}
```

will list ONLY files containing the word "picasso" or "Picasso" or "PICASSO" etc. You get the idea! Embedded spaces are NOT allowed, sorry.

1.47 freeware

By being FREEWARE, the terms of use are simple. You can use this program for as long as you like without having to pay ANY money. CFX falls under a slightly different category: REGISTERABLE FREEWARE. This means that you CAN in fact register CFX, and remain on the author's mailing list. Any time you want a new CFX (or any of Bob's other products) all you need to do is send a blank disk to Bob, along with appropriate prepaid return postage, and you can expect to see your new version(s) of the requested programs very soon. Of course, you can register in the standard way (ie. \$A25) but you could also register by sending Bob a registered copy of your program, or a paperback novel (sf or fantasy), or a cd (anything) or a spare Kelly Bundy or two...