



The
Graham Utilities
for
OS/2

*WarpSpeed*tm
Computers

Published by:

WarpSpeed Computers
PO Box 212
Brunswick VIC 3056
Australia

Copyright © 1994, 1995, 1996 by Chris Graham

All rights reserved. No part of this publication may be stored in a retrieval system, transmitted, or reproduced in any way, including but not limited to, photocopy, photograph, or magnetic or other record, without the prior written permission of the publisher.

Printed and bound by IBM SMS, Sydney, Australia

Eighth edition December 1996

Program Version	2.00
Manual Version	2.00

ISBN 0 646 18082 7

WARPSPEED COMPUTERS PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This publication could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. WarpSpeed Computers may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time.

All trademarks are the property of their respective holders.

Acknowledgments

The author would like to thank the following people; without their help this product would not have been possible:

Alan Williamson, Craig Smith, David Begley, David Nugent, David Sealey, David Wilkins, Mark Ames and Rainer Stober for testing and reading.

Debra Billson for the artwork ideas.

Gerry Rozema and Peter Fitzsimmons for lots of ideas and programming support.

Hermann Reissig for the translation into German.

Michael K. Vance for running the US based Web site and Tech Support.

Allan Mertner, Harry Bush and Doug Azzarito for their help with some HPFS information.

To all of those of you on Fidonet, CompuServe and Internet who have helped.

A very loud thank you to: Ozzy Osbourne, Steve Vai, Joe Satriani, Stuart Hamm, Metallica, Iron Maiden, Van Halen, Black Sabbath, Yngwie J. Malmsteen, Judas Priest et. al.

Sean Erwin for all of his help with sanity, a place to stay and the web page authoring.

And a very, very special thank you to Brian Post, for some last minute programming help and support. Without his help you would probably be still waiting for LD and GCD to build their directory trees!

Another special thank you to Eberhard Mattes who has helped by looking over my code and making many constructive criticisms and comments, and taught me a thing or two in the process!

Especially to Kate.

I am indebted to you all.

Contents

Contents	v
Introduction	1
Dedication	1
Overview	1
About the Utilities	3
OS/2	3
Switches	3
HELP	5
Dialog Boxes	6
BE - Batch Enhancer	7
BE ASK	9
BE BEEP	11
BE BOX	13
BE CLS	15
BE DAY	16
BE DELAY	17
BE MONTH	18
BE PRINTCHAR	19
BE REBOOT	20
BE ROWCOL	22
BE SA	24
BE SEMCLEAR	27
BE SEMCLOSE	28
BE SEMCREATE	29
BE SEMKILL	30
BE SEMLIST	31
BE SEMSET	32
BE SEMWAIT	33
BE WEEKDAY	35
BE WINDOW	36
BE YEAR	38
DI - Disk Information	39
-a All disks	42
-h Hard disks only	42
-p Physical disk mode	43
Networks	44
DS - Dir Sort	45
-a All disks	46

-h	Hard disks only	46
-s	Subdirectories	46
-on	Run on <Machine Name>	47
DT - Disk Test		49
-a	All disks	51
-b	Both File and Logical disk mode	51
-c	Cylinder mode	51
-d	Logical disk mode	51
-h	Hard disks only	51
-l	Keep a Log file	51
-p	Physical disk mode	52
-r	ReRun count	52
-s	Subdirectories	52
FA - File Attributes		55
+/-r	Read-Only Attribute	56
+/-a	Archive Attribute	56
+/-s	System Attribute	56
+/-h	Hidden Attribute	56
/a	All disks	56
/f	Force file names to lower case	57
/h	Hard disks only	57
/s	Subdirectories	57
FD - File Date		59
-a	All disks	60
-d	Specify the date	60
-f	Force file names to lower case	60
-h	Hard disks only	60
-s	Subdirectories	60
-t	Specify the time	61
FF - File Find		63
-a	All disks	63
-d	Delete found files	64
-f	Force file names to lower case	64
-h	Hard disks only	64
FI - File Information		65
-a	All disks	66
-c	Only files with comments attached	66
-d	Delete comments	66
-e	Edit or Add comments interactively	66
-h	Hard disks only	66
-l	Long comment listing	66
-n	Set EA as needed	67

-s	Subdirectories	67
-w	Only files without comments attached	67
FS -	File Size	69
-a	All disks	69
-f	Force file names to lower case	70
-h	Hard disks only	70
-s	Subdirectories	70
-t	Totals Only	70
GCD -	Graham Change Directories	73
-i	Use path to find GCD.INI	76
-n	Do NOT save directory data	76
-r	Rescan the specified disk	76
GI -	Graham Integrator	79
LD -	List Directories	81
-a	All disks	81
-g	Produce a Graphic directory tree	82
-h	Hard disks only	83
SA -	Screen Attributes	85
SI -	System Information	87
TM -	Time Mark	91
-c	Start or Stop timer	92
-l	Write on the left side of screen	92
-n	No current time or date display	92
VL -	Volume Label	93
WC -	Word Count	95
-a	All disks	95
-h	Hard disks only	96
-s	Subdirectories	96
2LZH -	Automatic archive conversion	97
-a	All disks	98
-c	Compact mode; keeps the smaller of the compressed files	98
-f	Modify Fidonet style FILES.BBS files	98
-h	Hard disks only	98
-i	Use path to find 2LZH.INI	98
-l	Keep a Log file	99
-n	No recursion	99
-o	Old (-lh1-) compatibility mode	99
-p	Preserve original archive time and date	99
-r	Repack *.LZH files	99
-s	List file Statistics	99
-u	Update mode; replace duplicates	100
-v	Virus scan unpacked files	100

FILES.BBS files	101
BEEP - Plays tunes	103
Options	103
Case - Change the case of files	105
-a All disks	106
-d Directories	106
-f First letter of each filename segment is upper	106
-h Hard disks only	106
-l Lower case file names	106
-s Subdirectories	106
-u Upper case file names	106
-w first letter of each Word is upper	107
Clipbrd - Copy text to the clipboard	109
-l Large Clipboard (>64K)	109
Config - Configure System Settings	111
System Configuration	112
Time Formats	113
Editing Options	113
Colour Options	114
-r Reset to default values	119
CSDInst - Install CSD Files	121
DIAG - System Diagnostics	123
System Configuration	124
Communication Port Tests	131
Video Tests	142
Hard Disk Tests	144
Floppy Disk Tests	144
Keyboard Tests	146
DiskEdit - Disk Editor	149
-nolock No disk lock is required	156
-allowwrites Allow disk writes	156
-c Specify an initial Cluster	157
-i Toggle the initial editing mode	157
-l Specify an initial Logical Sector	157
-p Physical disk editing mode	157
-r Force RAW disk editing mode	157
-u Update HPFS386 check sums	157
-on Run on <Machine Name>	158
DiskImg - Disk Imaging Tool	169
-r Read from the disk	173
-q Quiet mode	173
-w Write to the disk	173

EABackup - Backup Extended Attributes	175
-a All disks	176
-d Delete Extended Attributes after archive	176
-h Hard disks only	176
-s Subdirectories	176
EABind - Automatically bind EA's to files	177
-n Set EA as needed	178
-s Subdirectories	178
EADump - Dump Extended Attributes of a file	181
Extended Attribute Data Format	182
Intel Byte Ordering	182
EAT_ Values	182
EA Flag Values	183
-a All disks	183
-h Hard disks only	183
-l List the extended attribute names only	183
-n Dump extended attributes of a given name.	183
-o Only list files with extended attributes	184
-r Dump EAT_MVMT and EAT_MVST data as raw hex	184
-s Subdirectories	184
EARestor - Restore Extended Attributes	187
FAT-Bad - Set bad clusters on FAT disks	189
-on Run on <Machine Name>	190
FAT-Dfrg - Defragment FAT files	191
-h Hard disks only	192
-o Specify optimisation threshold	192
-s Subdirectories	192
-on Run on <Machine Name>	193
FAT-Ext - Report on FAT file fragmentation	195
-e Extents first	196
-h Hard disks only	196
-s Subdirectories	196
-on Run on <Machine Name>	196
FAT-Info - Report on FAT internals	199
-b Display bad sector lists	200
-f Display the free spaces	200
-h Hard disks only	200
-on Run on <Machine Name>	201
FAT-Null - Wipe FAT disks	203
-g Government security rules for wiping	204
-r Repeat wipe count	204
-v Wipe Value	204

-on Run on <Machine Name>	204
FAT-UD - Undelete files on FAT drives	205
-l Lock the drive	207
-p Prompt for restore path	207
-s Subdirectories are created	207
-on Run on <Machine Name>	207
FAT-View - View FAT drives	213
-c Clear the free clusters after defragment	216
-d Defragment files	216
-g Government security rules for wiping	216
-n Number of buffers; default of 100	216
-r Repeat wipe count	217
-u Unattended mode	217
-v Wipe Value	217
-on Run on <Machine Name>	217
FromUNIX - UNIX to OS/2 text conversion utility	221
GREP - Text search utility	223
-a All disks	224
-c Show the matched characters in Colour	224
-f List the File name	224
-h Hard disks only	224
-i Make the pattern case Insensitive	224
-n List the line Number	224
-o Only list the file name	224
-s Subdirectories	225
-x List the lines that do Not have pattern in them	225
Hexdump - Dump a file in Hexadecimal format	227
-a All disks	227
-c Clip the high bit off	228
-h Hard disks only	228
-s Subdirectories	228
-t Open the file in TEXT mode	228
-x Xor the high bit	228
HexEdit - Edit a file in Hexadecimal format	231
HPFS-Bad - Set bad sectors on HPFS disks	237
-on Run on <Machine Name>	238
HPFS-Ext - Report on HPFS file fragmentation	239
-e Extents first	240
-f Force file names to lower case	240
-h Hard disks only	240
-s Subdirectories	240
-on Run on <Machine Name>	240

HPFS-UD - Undelete files on HPFS drives	243
-c Complete disk scan	245
-d Damaged partition mode	246
-f Keep a log file	246
-l Lock the drive	246
-p Prompt for restore path	246
-s Subdirectories are created	246
-z Full surface scan for root dir FNode	247
-on Run on <Machine Name>	247
HPFSDfrg - Defragment HPFS files	253
-d Display defragged files only	254
-f Force file names to lower case	254
-h Hard disks only	254
-o Specify optimisation threshold	255
-r Retry Count	255
-s Subdirectories	255
-on Run on <Machine Name>	255
HPFSInfo - Report on HPFS internals	257
-b Display bad sector lists	260
-c Display the spare directory clusters	260
-d Display dynamic hot fix list	260
-f Display the free spaces	260
-h Hard disks only	261
-m Display the bitmap sector lists	261
-on Run on <Machine Name>	261
HPFSNull - Wipe HPFS disks	263
-g Government security rules for wiping	264
-r Repeat wipe count	264
-v Wipe Value	264
-on Run on <Machine Name>	264
HPFSStat - Examine the status of HPFS disks	265
-c Mark the partition as CLEAN	266
-d Mark the partition as DIRTY	266
-r Remount the partition	266
-on Run on <Machine Name>	266
HPFSView - View HPFS drives	269
-d Defragment files	272
-o Specify optimisation threshold	272
-on Run on <Machine Name>	272
MIDdecode - Decode MIME files	275
Possible Decoding Problems	276
-e decode Extended Attribute data	276

MIEncode - Encode MIME files	277
-c Console output	278
-e encode Extended Attribute data	278
-s include SUM information in the file	279
-t include TABLE information in the file	279
-u UNIX compatible file	279
NullDisk - Wipe Disk data	281
-a All disks	282
-e Erase unused and deleted file areas only	282
-g Government security rules for wiping	282
-h Hard disks only	282
-n No wipe, erase only	282
-r Repeat wipe count	283
-v Wipe Value	283
NullFile - Wipe File data	285
-a All disks	286
-d Remove directories	286
-g Government security rules for wiping	286
-h Hard disks only	286
-n No wipe, erase only	286
-p Prompt for wiping or deleting	286
-r Repeat wipe count	286
-s Subdirectories	287
-v wipe Value	287
RestPart - Restore saved partition information	289
SavePart - Save partition information	295
-v Verbose mode	296
Space - Disk Drive Space	297
-c Colour warning display	298
-g Graphic representation	298
-p Pause between graphic screens	298
SUM - Check Sum Files	301
-a All disks	302
-h Hard disks only	302
-s Subdirectories	302
ToUNIX - OS/2 to UNIX text conversion utility	305
UUDecode - Decode UU files	307
-e decode Extended Attribute data	308
-q do not use Quote character (") instead of space	308
UUEncode - Encode UU files	309
-c Console output	310
-e encode Extended Attribute data	310

-q	do not use Quote character (") instead of space	311
-s	include SUM information in the file	311
-t	include TABLE information in the file	311
-u	UNIX compatible file	311
XXDecode	- Decode XX files	313
-e	decode Extended Attribute data	314
-q	do not use Quote character (") instead of space	314
XXEncode	- Encode XX files	315
-c	Console output	316
-e	encode Extended Attribute data	316
-q	do not use Quote character (") instead of space	317
-s	include SUM information in the file	317
-t	include TABLE information in the file	317
-u	UNIX compatible file	317
Appendix A	- Extended Attributes	319
Extended Attributes	- Overview	319
Extended Attributes	- Details	320
Extended Attributes	- Usage	321
Appendix B	- SEMMGR	323
Appendix C	- 2LZH.INI	327
Archives Section		327
Virus Section		328
2LZH.INI		328
Appendix D	- WARP.INI	331
Appendix E	- Extra Utilities	333
Appendix F	- CompuServe	337
Appendix G	- HPFS Internals	347
Boot Blocks		347
Super Block		348
Spare Block		350
FNodes		352
ALSECs		354
DIRBLKs		355
Extended Attributes and ACLs		356
Appendix H	- DESVR	357
Appendix I	- DiskEdit Modules	359
Edit Partition Table		359
Edit Boot Sector		364
Edit FAT		367
Edit Directory		370
Edit Raw Hex		372
Edit Super Block		372

Edit Spare Block	374
Edit Bitmap	376
Edit Directory Bitmap	377
Edit DirBlk	377
Edit FNODE	381
Edit ALSEC	384
Edit Code Page Information	386
Edit Code Page	387
Edit Bad Block List	390
Edit Hot Fix List	391
Edit Bitmap List	392
Appendix J - Internet Services	393
Available Services	394
WWW	394
FTP	395
GOPHER	395
TELNET	395
UPDATES	395
Manually installing a CSD file	396
Other sites	396
Glossary	397
License Agreement	408

Introduction

Dedication

This collection of programs is dedicated to the memory of Randy Rhoads (1956-1982). Thank you for much inspiration. You will not be forgotten.

Overview

Welcome to the Graham Utilities for OS/2. They are a comprehensive set of file, disk and general utilities specifically written for the OS/2 operating system. They have been designed to exploit the power of OS/2 and enable you to be more efficient in using OS/2.

I became acquainted with OS/2 in 1989, using OS/2 1.2. At this point in time it was hard to find many useful OS/2 utilities. You could use the DOS box for some utilities, but most things needed to be OS/2 specific. So, with that, and the habit of typing TM and FF, I, like a few of my friends, was forced to write my own utilities. Over time, the utilities grew and grew. The result is what you see before you. I hope that you enjoy these utilities and find them useful.

If you have any comments, questions or suggestions for future enhancements or additions, please feel free to contact me.



31st August 1996

Postal Mail:

WarpSpeed Computers
PO Box 212
Brunswick VIC 3056
AUSTRALIA

Phone details:

Phone: +61-3-9384-1060
Fax: +61-3-9386-9979
BBS: +61-3-9386-3104
300 - 28,800 8,N,1 ANSI

Please note: Australia is 14 - 16 hours AHEAD of US east coast time.

EMail:

FidoNet: 3:632/344.0
Internet: 100250.1645@CompuServe.Com
CompuServe: 100250,1645

Support:

Support is offered via Phone, Fax, Letter and EMail. EMail is the preferred method. The primary means of electronic support is through CompuServe. Questions and comments may be left in the Other Vendors section (section 1) of the OS2AVEN forum.

Support is also available in the US for Canada and the US.

Postal Mail:

WarpSpeed Computers
15172 25th Street
Morning Sun IA 52640-9403
USA

Phone details:

Phone: +1-319-868-7656
Fax: +1-319-868-7656
Phone: +1-800-644-5669
(Canada & USA)

About the Utilities

OS/2

All of "The Graham Utilities for OS/2" are OS/2 protected mode applications; they will NOT run in the DOS Compatibility box. All are capable of being run in either an OS/2 Windowed session (known as a VIO window) or an OS/2 Full Screen session, with one exception. The exception is DIAG which can only be run in a Full Screen session. Some of the applications, such as HEXEDIT, which use a popup window will automatically size themselves to use the full dimensions of the current screen mode. The screen mode is set using the MODE command. The most information which you can display on a standard VGA screen (in Full Screen mode) is 80 columns by 60 rows, which can be set by the command:

```
MODE CO80,60
```

Some display adaptors support 132 column modes. The maximum number of rows for 132 column modes is 44. This can be set by the command:

```
MODE CO132,44
```

Version two of The Graham Utilities for OS/2 supports all versions of OS/2. There are 16 bit applications that are capable of running under IBM or Microsoft OS/2 1.2, OS/2 1.3 and IBM OS/2 2.x, 3.x (OS/2 Warp), 4.x (Merlin) and Microsoft Windows NT. There are also 32 bit applications that will only run under IBM OS/2 2.x, 3.x and 4.x. Also included in this release is a version of the applications which will run under OS/2 Warp for the Power PC.

Note:

Some of the 16 bit programs will not successfully run under Microsoft Windows NT. These programs are ones which perform direct disk accesses, and the BE REBOOT command. None of these programs will run correctly due to the security provided by Microsoft Windows NT.

Switches

All of the utilities in "The Graham Utilities for OS/2" follow the same convention for parsing command line switches. A switch is a command which is placed on the command line of a program to change its functionality. All of the

utilities use a switch character (a "-" or "/") followed by a single letter. For example:

```
DI -p
```

changes the functionality of DI from its default mode of displaying information about logical disk partitions to displaying information about the physical disk(s) attached to your system.

If a program supports multiple switches, the switches may be concatenated together. They do not have to be individually specified. For example:

```
LD /g /h
```

is identical to:

```
LD /gh
```

There is one exception to the concatenation rule: when an argument for the switch has to be given. The argument must be specified by itself or be the last specified. An example of this is NullFile; some of the switches for NullFile require a numeric argument. The help display lists the switch in the format:

```
-v<n> wipe Value; default of 0.
```

The angled brackets (" $<$ " and " $>$ ") are not actually typed: they indicate that a number should be entered. This is an example of how it could be entered at the command prompt:

```
NULLFILE -v85 -ns
```

Note that there is no space between the "-v" and "85".

```
NULLFILE -v85ns
```

This is illegal, the -n and -s switches will be ignored. It could be specified as:

```
NULLFILE -v85 -ns
```

or

```
NULLFILE -nsv85
```

Neither the order nor the case of the switches are important, so the following are all identical:

```
GREP -F -I -N -C main *.c
GREP -FINC main *.c
GREP -finc main *.c
GREP /finc main *.c
GREP -fn main *.c /ic
```

HELP

All of the utilities have their own on-line help. This will display a brief summary of all of the available switches and the arguments required for each program.

The format of the help is as follows:

Switches	Are listed in between curly braces - '{' and '}'. With the exception of numeric arguments they can be entered as displayed.
Options	Are listed in between square brackets - '[' and ']'. They are key words and must be entered exactly as shown. Case is not important.
Parameters	Are listed in between angled brackets - '<' and '>'. They indicate that a parameter must be entered, but you supply the parameter. Typically a file specification is an example of a parameter. Parameters are optional. If they are not given, the program will continue with default values.

To see the quick help simply enter the program name followed by a "-?" or "/"? switch. For example:

```
HEXDUMP -?
```

In the event that the help switch is issued in conjunction with any other switches, the program will display its help and then terminate with an exit code of 1.

If any unrecognised switches are issued then the program will report it as such, display the help, and terminate with an exit code of 1.

If the program runs successfully, it will return an error code of 0, unless otherwise specified.

A more comprehensive form of on line help is also provided in the GI (Graham Integrator) program. GI will display a list of all programs, their options, command line arguments, a brief program description, and sometimes, an example of usage.

The most comprehensive form of help is offered in the on-line manual GRAHAM20.INF. You may use the standard OS/2 command, VIEW to examine the manual.

```
VIEW GRAHAM20
```

This will enable you to view the on-line manual.

GI also enables you to view the on-line manual. Highlight the program for which you wish to view the detailed help and press the F2 key. You will be presented with the chapter on the requested program.

Dialog Boxes

Some of the programs use popup windows and dialog boxes to display information and to prompt you for action. When a dialog box appears, you may generally use the following keys:

Enter	Select the currently highlighted item and continue.
Tab	Move from one field to another.
Shift Tab	Reverse the move from one field to another.
Arrow Keys	Move in the direction indicated. At times only the left and right arrows will be functional.
Escape	Exit the program or dialog box. No action taken.
Highlighted Letters	Press the letter to action that item.

In some dialog boxes, such as the Select Drive dialog box, just pressing the letter is all that is required of you.

BE - Batch Enhancer

Summary

BE may be used to enhance batch files (.CMD). It has various sub commands available. Use BE to create interactive, more powerful batch files.

Icon



Command Line Format

Usage: BE <File Spec>
 or
 BE <command> <parameters>

Where <command> is one of the following:

ASK	BEEP	BOX	CLS
DAY	DELAY	MONTH	PRINTCHAR
REBOOT	ROWCOL	SA	SEMCLEAR
SEMCLOSE	SEMCREATE	SEMKILL	SEMLIST
SEMSET	SEMWAIT	WEEKDAY	WINDOW
YEAR			

Enter BE <command> ? to get help on a specific command.

Description

BE has two command formats. It can be passed all of its command line arguments at once, for example:

```
BE SA White on Blue
```

Or BE can be passed a response file to parse. In this mode BE can be passed multiple commands all at once, and they will be executed in sequence. Be aware however, that some commands (SEMWAIT for example) will return an exit code. The response file will not be processed beyond any command which

returns an exit code. Return codes are useful for a batch programmer to enable the control of the execution of a .CMD file.

A response file is a file which contains BE commands and parameters for batch processing. Each line in the file contains one BE command. Response files do not require BE to be on each individual line. For example:

```
BE TestFile
```

where TestFile contains:

```
SA White on Blue
```

All of the following examples will be of the BE <command> <parameters> format.

BE ASK

Summary

BE ASK prompts the user for a response to a question. Use ASK to make conditional branches (jumps) in your .CMD files.

Command Line Format

Usage: BE ASK "prompt" key-list <DEFAULT=key><TIMEOUT=n>
<ADJUST=n> <Colour Spec>

Parameters

"prompt "	The text string with which the user is presented. If it is more than a single word, it must be surrounded by double quotes (").
key-list	Key-list is the list of valid responses that the user may select. The first character will return an error level of 1, the second an error level of 2 and so on.
DEFAULT=key	If the user presses Enter or uses it with the TIMEOUT option, ASK will behave as if the user had selected this response. This is an optional parameter. If DEFAULT is not specified, BE will return an error level of 255.
TIMEOUT=n	Used to return an answer after n seconds if the user has not pressed a key. This should be used with the DEFAULT option. This is an optional parameter. If it is not specified, ASK will continue to wait until a valid key is pressed.
ADJUST=n	Adjust allows you to specify an offset to the error code returned to the OS/2 command processor. The value of n will be added to all return values. This is an optional parameter.
Colour Spec	Colour Spec is a colour specification. See BE SA for further details. This is an optional parameter.

Description

BE ASK prompts the user for a response to a question. ASK will return an error code to OS/2 so that the IF ERRORLEVEL batch command can be used to make conditional branches (jumps) in your .CMD files.

Note that both the "prompt" and [key-list] options must be supplied.

Examples

```
Rem this is the first sample batch file
BE ASK "Start the network in Server mode [Y/n]?" YN Default=Y
timeout=10
If errorlevel 2 goto Requester
Rem Start the network in server mode
Net Start Server
Goto Quit
:Requester
BE ASK "Start the network in Requester mode [Y/n]?" YN
Default=Y timeout=10
If errorlevel 2 goto Quit
Net Start Requester
:Quit
```

This batch file will ask the user if they wish to start IBM LAN Server in Server mode. If the user responds "No" to this, by pressing the "N" key, they will be further prompted to start LAN Server in Requester mode. If the user does not press any key within 10 seconds, the default option of Y is used to start the network in server mode.

```
REM this is the second sample batch file
BE ASK "Press the Y key only" Y
Echo You pressed the Y key
:Quit
```

This example is as about as simple as you can get. It prompts the user to only press the 'Y' key.

BE BEEP

Summary

BE BEEP plays a tune file.

Command Line Format

Usage: BE BEEP <file>

Options

The following options are valid in tune files:

/Fn	Specify the frequency of the tone in Hertz.
/Dn	Specify the duration of the tone in n/18 seconds.
/Tn	Specify the duration of the tone in milliseconds.
/Rn	Repeat a tone n times.
/Wn	Wait n/18 seconds between tones.

They can also be specified in raw Frequency Duration pairs. This makes BEEP compatible with Maximus BBS format tune files.

The repeat and wait options only apply to the line on which they appear.

Tune files may also have comments in them. Comments start with a semicolon ";".

Description

BE BEEP allows you to play tunes on your computer. It supports a wide variety of formats.

Two sample tune files have been provided. These are INDIANA and SIMPSONS.

Examples

```
BE BEEP INDIANA
```

Will play the "Indiana Jones" theme.

BE BEEP SIMPSONS

Will play the theme from "The Simpsons".

BE BOX

Summary

BOX allows you to display a box on your text screen.

Command Line Format

Usage: BE BOX top left bottom right <colour spec> <STYLE=n>

Parameters

top	Specifies the row position of the top left hand corner of the box.
left	Specifies the column position of the top left hand corner of the box.
bottom	Specifies the row position of the bottom right hand corner of the box.
right	Specifies the column position of the bottom right hand corner of the box.
colour spec	Specifies a colour specification. See BE SA for further details. This is an optional parameter.
STYLE=n	This allows you to specify the style of the box which is drawn. Valid values for n are: 0 No border is drawn. 1 A single line border is drawn. 2 A double line border is drawn. The default style is 2.

The row and column coordinates can range from one up to the maximum row and column coordinates as specified by the MODE command. These values have a default value which is dependent on your video adaptor.

Description

BOX allows you to display a box on the screen. Only the boarder of the box is drawn. The body, or centre of the box is left as is. The type of box characters drawn can be specified by the STYLE parameter. A SA colour specification may also be added to set the colours of the drawn box.

Examples

```
BE BOX 1 1 25 80
```

In standard VGA screen mode (80 columns by 25 rows), the above command will display a box which will outline the entire screen in the current screen colours.

```
BE BOX 10 10 15 70 Bright Green on Black
```

This will display a box using bright green characters on a black background in the middle of the screen.

```
BE BOX 10 10 15 70 Bright Red on Black Style=1
```

This will display a box using bright red characters on a black background in the middle of the screen. The box will be drawn using single line box characters.

BE CLS

Summary

Clears the current screen.

Command Line Format

Usage: BE CLS

Description

BE CLS will clear the current screen. This is identical to the OS/2 CLS command and is included so that it may be used in a response file.

Example

```
BE CLS
```

This will clear the current screen. The current screen colour remains unchanged.

BE DAY

Summary

The BE DAY returns the current day of the month as an error level.

Command Line Format

Usage: BE DAY

Description

BE DAY will return the current day of the month as an error level so that batch programs can carry out actions on specified days. This BE sub command will return an error level to the OS/2 command interpreter. With the use of this command any processing of a BE response file will cease.

Example

BE DAY

The current day of the month will be returned as an error level. The returned values are in the range of 1 to 31.

BE DELAY

Summary

BE DELAY waits for a specified period.

Command Line Format

Usage: BE DELAY ticks {-ESC} (1 tick = 1/18 second)

Parameters

ticks	Specifies the time period of the delay. Each tick is 1/18 of a second or approximately 55 mSec.
-ESC	Allows early escape from the delay by pressing the Escape key.

Description

The BE DELAY command is useful to force a batch file to pause for a fixed period of time. You could use the BE DELAY command to allow some of your computers on a network to boot and establish themselves before they start to share each other's resources.

Examples

```
Net Start Server
REM Wait for 5 minutes
REM Wait for the other machine to boot and wait for us
BE DELAY 5400
Logon Chris
Net Use H: Main-H
```

This example starts LAN Server in server mode. BE then waits for 5 minutes for another server to boot and then logs on and uses the resources of other machines.

BE MONTH

Summary

BE MONTH returns the current month of the year as an error level.

Command Line Format

Usage: BE MONTH

Description

BE MONTH will return the current month of the year as an error level to enable batch programs to carry out actions in a specified month. This BE sub command will return an error level to the OS/2 command interpreter. Any processing of a BE response file will cease with this command. The month of January will return an error level of 1 and so on.

Example

```
BE MONTH
```

The current month of the year will be returned as an error level. The returned values are in the range of 1 to 12.

BE PRINTCHAR

Summary

BE PRINTCHAR prints a specified character at the current cursor position a defined number of times.

Command Line Format

Usage: BE PRINTCHAR character count <colour spec>

Parameters

character	Specifies the character to be printed.
count	Specifies the number of times the character is to be printed.
colour spec	Specifies a colour specification. See BE SA for further details. This is an optional parameter.

Description

BE PRINTCHAR is a convenient way of displaying a single character multiple times. Using BE ROWCOL will establish the cursor position.

Note:

If more than one character is given, then only the first one will be printed.

Examples

```
BE PRINTCHAR A 20 bright green on blue
```

This will print the letter "A" 20 times at the current cursor position. The "A" will be bright green on a blue background.

```
BE PRINTCHAR " " 10
```

This example will print 10 spaces at the current cursor position.

BE REBOOT

Summary

BE REBOOT will shutdown all file systems and then reboot your computer.

Command Line Format

Usage: BE REBOOT { -NOVERIFY }

Switches

-NOVERIFY Specifies that the user will NOT be prompted to confirm the reboot of the computer.

Description

BE REBOOT allows you to reboot your computer. It will shutdown all file systems, close all open files and then reboot.

WARNING:

USING THIS METHOD TO REBOOT YOUR COMPUTER IS DANGEROUS. THE CURRENTLY ACTIVE PROGRAMS RECEIVE NO WARNING THAT THE MACHINE IS ABOUT TO BE REBOOTED. DATA LOSS COULD RESULT BECAUSE A PROGRAM MAY HAVE DATA IN MEMORY WHICH HAS NOT BEEN SAVED TO DISK.

This option of BE requires the DOS compatibility box to be installed. If BE is unable to reboot the computer due to the DOS compatibility box not being installed you will be given the following error message:

```
DOS.SYS not installed (SYS0110).
```

Microsoft Windows NT does not support this command. Using Windows NT you will see the above error.

Examples

```
BE REBOOT
```

With this example you will be asked if you wish to continue and reboot the computer, or cancel the command.

```
BE REBOOT /NOVERIFY
```

This example will not prompt the user, it will simply reboot the computer.

It is recommended that this command is only used when a user will not be in attendance. An example could be to force a machine to reboot at midnight.

BE ROWCOL

Summary

BE ROWCOL will position the cursor at the specified location, and if required, print some text in a given colour.

Command Line Format

Usage: BE ROWCOL row col <"text"> <colour spec>

Parameters

row	Specifies the row position of the cursor.
col	Specifies the column position of the cursor.
"text"	Specifies the text to be displayed. This is an optional parameter.
colour spec	Colour is a colour specification. See BE SA for further details. This is an optional parameter.

Description

BE ROWCOL positions the cursor at the specified row and column position. It may also optionally display some text. The colour of the displayed text may also be specified. This option is most useful in a BE response file.

The row and column coordinates start at 1 and range up to the maximum row and column coordinates as specified by the MODE command. These values have a default value dependent on your video adaptor.

Examples

```
BE ROWCOL 10 10
```

Positions the cursor at 10, 10.

```
BE ROWCOL 15 40 Hello
```

Prints "Hello" at 15, 40.

```
BE ROWCOL 16 40 "Hello There"
```

Prints "Hello There" at 16,40.

```
BE ROWCOL 17 40 "Hello There Again" Bright Red
```

Prints "Hello There Again" at 17, 40 in bright red on a black background.

BE SA

Summary

BE SA allows the user to specify the current screen attributes, or colours.

Command Line Format

Usage: BE SA [bright] [blinking] [reverse] [underline]
<colour> <on colour> -c

Parameters

bright	Specifies that the foreground colour will be bright. This is an optional parameter.
blinking	Specifies that the displayed characters will blink or flash. This is an optional parameter.
reverse	Specifies that the text to be displayed will be displayed in black on white, unless over ridden by a foreground/background combination. This is an optional parameter.
underline	Specifies that the text to be displayed will be underlined. This will only work on monochrome monitors. Colour monitors will display any underlined text in blue.
colour	Specifies the foreground colour. It can be any one of the valid colours. This is an optional parameter.
on colour	Specifies the background colour. It can be any one of the valid colours. This is an optional parameter.
-c	This switch will clear the screen before the new colour attributes are set.

The valid colours are:

Black
Red
Green
Yellow
Blue
Magenta
Cyan
White

Description

BE SA allows the user to specify the current screen attributes, or colours. It sets the current screen attributes so that any further displayed text will appear in those attributes unless overridden with a colour specification from another BE SA or similar command.

Examples

```
BE SA RED
```

This will set the current foreground colour to red. The background colour is set to black.

```
BE SA ON RED
```

This will set the current background colour to white on red. If the foreground colour is not specified it will default to white.

```
BE SA Bright green
```

This will set the foreground colour to bright green on black.

```
BE SA REVERSE -c
```

This will set the current text colours to black text on a white background, after the screen has been cleared.

```
BE SA blue reverse
```

This will set the current text colours to blue on a white background.

```
BE SA blinking red
```

This will set the current foreground text colour to blinking red on a black background.

BE SEMCLEAR

Summary

BE SEMCLEAR is used to clear a system semaphore.

Command Line Format

Usage: BE SEMCLEAR <semaphore>

Parameter

semaphore This is the semaphore which SEMMGR will clear. It must be a valid OS/2 file name.

Description

A system semaphore has two states; set or cleared. The BE SEMCLEAR command clears the specified semaphore. Please consult Appendix B - SEMMGR for a description of SEMMGR and semaphores.

Examples

```
BE SEMCLEAR test
```

This example clears the semaphore \SEM\SEMMGR\TEST. It must have been previously created using the BE SEMCREATE command.

```
BE SEMCLEAR test\test
```

This example clears the semaphore \SEM\SEMMGR\TEST\TEST.

BE SEMCLOSE

Summary

BE SEMCLOSE closes a system semaphore.

Command Line Format

Usage: BE SEMCLOSE <semaphore>

Parameter

semaphore This is the semaphore which SEMMGR will close. It must be a valid OS/2 file name.

Description

The BE SEMCLOSE command closes a semaphore which is managed by SEMMGR. The semaphore must be cleared before SEMCLOSE can be called. If it is not cleared, an error will occur. When the semaphore is closed, it is no longer maintained by SEMMGR. Thus it no longer exists. For the same semaphore to be reused again, it must be recreated with the BE SEMCREATE command. Please consult Appendix B - SEMMGR for a description of SEMMGR and semaphores.

Examples

```
BE SEMCLOSE test
```

This example closes the semaphore \SEM\SEMMGR\test.

```
BE SEMCLOSE test\test
```

This example closes the semaphore \SEM\SEMMGR\test\test.

BE SEMCREATE

Summary

BE SEMCREATE is used to create a system semaphore.

Command Line Format

Usage: BE SEMCREATE <semaphore>

Parameters

semaphore This is the semaphore which SEMMGR will create. It must be a valid OS/2 file name.

Description

A semaphore must be created before it can be used. BE SEMCREATE does this. When a semaphore is created, its initial state is cleared. See Appendix B for a complete discussion on the use of semaphores.

SEMMGR has a maximum limit of 128 semaphores.

Examples

```
BE SEMCREATE test
```

This example will create a semaphore. The complete semaphore name is \SEM\SEMMGR\test. However you will only supply the name which follows \SEM\SEMMGR\. SEMMGR builds up the complete semaphore name for you.

```
BE SEMCREATE test\test
```

The semaphore \SEM\SEMMGR\TEST\TEST will be created for you. It may then be used in subsequent semaphore operations.

BE SEMKILL

Summary

BE SEMKILL terminates the detached process SEMMGR.

Command Line Format

Usage: BE SEMKILL

Description

SEMMGR is a detached process. It runs in the background. It has no way of accepting commands from the user, except from BE. The BE SEMKILL command sends a command to SEMMGR to force SEMMGR to terminate.

WARNING

When SEMMGR is terminated it clears all semaphores and closes them. This may cause unexpected results in your batch programs. Its use is not recommended, although sometimes it is necessary to unilaterally clear and close all SEMMGR managed semaphores. Please consult Appendix B - SEMMGR for a description of SEMMGR and semaphores.

Example

```
BE SEMKILL
```

The BE SEMKILL command has no parameters. This is the only way in which it may be called.

BE SEMLIST

Summary

BE SEMLIST lists all of the semaphores currently managed by SEMMGR and their state.

Command Line Format

Usage: BE SEMLIST

Description

The BE SEMLIST command lists the full name of each of the semaphores managed by SEMMGR as well as their state. Please consult Appendix B - SEMMGR for a description of SEMMGR and semaphores.

Example

```
BE SEMLIST
```

There are no parameters for the BE SEMLIST command. The output of the command will look similar to:

```
State   Semaphore Name
-----  -
Set     \SEM\SEMMGR\test
Clear   \SEM\SEMMGR\test\test
```

BE SEMSET

Summary

BE SEMSET is used to clear a system semaphore.

Command Line Format

Usage: BE SEMSET <semaphore>

Parameter

semaphore This is the semaphore which SEMMGR will set. It must be a valid OS/2 file name.

Description

A system semaphore has two states; set or cleared. The BE SEMSET command sets the specified semaphore. The BE SEMSET command waits for the semaphore to be cleared before it is again set. This is to stop semaphore contention problems. Please consult Appendix B - SEMMGR for a description of SEMMGR and semaphores.

Examples

```
BE SEMSET test
```

This example sets the semaphore \SEM\SEMMGR\TEST. It must have been previously created using the BE SEMCREATE command.

```
BE SEMSET test\test
```

This example sets the semaphore \SEM\SEMMGR\TEST\TEST.

BE SEMWAIT

Summary

BE SEMWAIT waits for a specified semaphore to clear.

Command Line Format

Usage: BE SEMWAIT <semaphore> [timeout (in mSec)]

Parameters

semaphore	This is the semaphore which SEMMGR will wait on. It must be a valid OS/2 file name.
timeout	This specifies the timeout value during which BE will wait for the semaphore to clear. If no value is specified, BE will wait indefinitely.

Description

Semaphores have two states - they are either set or cleared. The default state of a semaphore when it is created is cleared. The BE SEMWAIT command waits for a semaphore's state to change from set to clear. A timeout period, (the period that you are prepared to wait for the semaphore to clear), may also be specified. The timeout period is specified in milliseconds. If a period is not given, BE will wait indefinitely. If the timeout period is given, and the semaphore has not cleared within that period, BE will return with an error level of 1. If the semaphore is already clear, or is cleared while BE is waiting for it, BE will return with an error level of 0. See Appendix B for a complete discussion on the use of SEMMGR and semaphores.

Unless it is the last command, this command should not be used in a response file. This BE sub command will return an error level to the OS/2 command interpreter. Any processing of a BE response file will cease with the use of this command.

Examples

```
BE SEMWAIT test
```

This example waits for the semaphore `\SEM\SEMMGR\TEST` to clear. BE will wait indefinitely for the semaphore to clear. It will return immediately if the semaphore is already clear. The error level returned is 0.

```
BE SEMWAIT test\test 10000
```

This example waits for the semaphore `\SEM\SEMMGR\TEST\TEST` to clear. If the semaphore has not cleared within the specified time (10 seconds), an error level of 1 will be returned. Otherwise the semaphore has cleared and an error level of 0 will be returned.

BE WEEKDAY

Summary

BE WEEKDAY returns the current day of the week as an error level.

Command Line Format

Usage: BE WEEKDAY

Description

BE WEEKDAY will return the current day of the week as an error level to enable batch programs to carry out actions on a specified day. This BE sub command will return an error level to the OS/2 command interpreter. Any processing of a BE response file will cease with this command. Sunday will return an error level of 1, Monday an error level of 2 and so on.

Example

```
BE WEEKDAY
```

The current day of the week will be returned as an error level. The returned values are in the range of 1 to 7.

BE WINDOW

Summary

BE WINDOW allows you to display a window on the text screen.

Command Line Format

Usage: BE WINDOW top left bottom right <colour spec> [EXPLODE]
[SHADOW] <STYLE=n>

Parameters

top	Specifies the row position of the top left hand corner of the box.
left	Specifies the column position of the top left hand corner of the box.
bottom	Specifies the row position of the bottom right hand corner of the box.
right	Specifies the column position of the bottom right hand corner of the box.
colour spec	Specifies a colour specification. See BE SA for further details. This is an optional parameter.
[EXPLODE]	Explodes the window. That is, the window is drawn in a number of stages. This is an optional parameter. This parameter is implemented in syntax only. It does not actually do anything.
[SHADOW]	Places a shadow under the window. This is an optional parameter. This parameter is implemented in syntax only. It does not actually do anything.

STYLE=n This allows you to specify the style of the box which is drawn.
Valid values for n are:
0 No border is drawn.
1 A single line border is drawn.
2 A double line border is drawn.
The default value is 2.

The row and column coordinates range from one up to the maximum row and column coordinates as specified by the **MODE** command. These values have a default value dependent on your video adaptor.

Description

BE WINDOW allows you to display a window on the screen. Unlike the **BE BOX** command, window fills in the body of the window with spaces, **BE BOX** does not. The type of box characters drawn can be specified by the **STYLE** parameter. A SA colour specification may also be added to set the colours of the drawn box.

Examples

```
BE WINDOW 1 1 25 80
```

In standard VGA screen mode (80 columns by 25 rows), the above command will display a window which will outline the entire screen in the current screen colours.

```
BE WINDOW 10 10 15 70 Bright Green on Black
```

This will display a bright green window in the middle of the screen.

BE YEAR

Summary

BE YEAR returns the current year as an error level.

Command Line Format

Usage: BE YEAR

Description

BE YEAR will return the current year as an error level to enable batch programs to carry out actions in a specified year. This BE sub command will return an error level to the OS/2 command interpreter. Any processing of a BE response file will cease with this command. The year 1995 will return an error level of 95 and so on.

Example

BE YEAR

The current year will be returned as an error level. The year 2000 will return a value of 100, the year 1994 an error level of 94 and so on.

DI - Disk Information

Summary

DI reports technical information about a disk. It displays the recommended values for the media as well as the actual values.

Icon



Command Line Format

Usage: DI {Switches} <Drive Spec(s)>

Switches

- a All disks.
- h Hard disks only.
- p Physical disk mode.

Note:

If no drive specifications are entered, the current drive is the only one displayed.

Description

DI reports on the technical information about a disk. OS/2 reports two different types of data about a disk - the recommended values and the actual values. The recommended values are those OS/2 thinks that particular type of media SHOULD have; and the actual values are those the media ACTUALLY reports. Normally these should be the same.

The values which DI reports on are:

bytes per sector	Sectors are the smallest amount of data which a disk can either read or write. They are determined when the disk is physically formatted. The bytes per sector value is usually 512, but it may be 128, 256, 1024 or 2048 as well.
sectors per cluster	OS/2 and DOS allocate sectors in groups called "Clusters". This is the minimum amount of space in which DOS or OS/2 can allocate to a file. This value varies, it depends on the size and type of media.
reserved sectors	This is the number of sectors reserved for use by the operating system.
number of FATs	The File Allocation Table (FAT) file system uses tables to save information about where files reside. To assist in maintaining file system integrity, more than one copy of the FAT may be kept. The default for the number of FAT's is 2.
root directory entries	The FAT file system requires a fixed space to be reserved for the root directory. This amount is determined when the disk is formatted. This value varies as it depends on the size and type of media. Although a value is returned for HPFS partitions, its value is meaningless as HPFS has no limit to the number of files in the root directory.
number of sectors	This specifies the number of sectors available on the disk. This value is only valid for partitions of less than 32Mb in size.

media descriptor	<p>Specifies a hexadecimal number which lists the type of media in the disk drive. Common values are:</p> <p>F0 A 1.44Mb 3.5" floppy disk F8 A hard disk. F9 A 1.2Mb 5.25" floppy disk OR A 720K 3.5" floppy disk. FD A 360K 5.25" floppy disk. FE A 160K 5.25" floppy disk. FF A 320K 5.25" floppy disk.</p>
sectors per FAT	Specifies the number of sectors occupied by each FAT in the FAT file system.
sectors per track	<p>Each track is divided up into a number of sectors. Common values are:</p> <p>8 A 320K 5.25" floppy disk. 9 A 360K 5.25" floppy disk OR A 720K 3.5" floppy disk. 15 A 1.2Mb 5.25" floppy disk. 18 A 1.44Mb floppy disk.</p>
number of heads	Specifies the number of heads (or sides) of the logical disk.
hidden sectors	Under FAT file systems, this specifies the number of sectors reserved for system use.
large sectors	This is the number of sectors available on a large partition. A large partition is > 32Mb in size.
number of cylinders	Specifies the number of cylinders (or tracks) on a hard drive or floppy disk.

device attributes Specifies the attributes of a particular device. They can be added together. The attributes are defined by the following values:

- 1 Not removable.
- 2 Can detect media changes.
- 4 The physical device driver supports physical addresses greater than 16 megabytes.

device type This specifies the type of device. The type of device is specified by the following:

- 0 48 tracks-per-inch, low density floppy disk.
- 1 96 tracks-per-inch, high density floppy disk.
- 2 3.5-inch (720K) floppy disk.
- 3 8-inch, single density floppy disk.
- 4 8-inch, double density floppy disk.
- 5 Fixed disk.
- 6 Tape drive.
- 7 Other or 1.44Mb (unknown type of device).
- 8 Read/Write Optical disk.
- 9 3.5-inch 2.88Mb floppy disk.

-a All disks

DI will scan all logical disk drives defined by the system, starting at drive A:. The drives are scanned in order, starting at A:, then B: and so on.

-h Hard disks only

DI will scan all logical disk drives defined by the system, starting at drive C:. The drives are scanned in order, starting at C:, then D: and so on.

-p Physical disk mode

By default, DI works in logical disk mode. Logical disks are defined by FDISK when you partition your hard disks. A logical disk must be part (or all) of a physical disk. Multiple logical disks can be defined on a single physical (or real) hard disk. The -p switch changes the operation of DI to enable it to report on information about the physical disks attached to your system. If this switch is given, then only the physical disk information is returned. No logical disk information is returned.

In this mode DI will report on the number of cylinders (tracks), heads, and sectors per track. The definitions of cylinders etc, are the same as described as above, but they apply to the physical disk, not a logical disk.

Examples

DI

DI will report on the information about the currently selected logical disk.

DI -h

DI will report the information about all of your logical disks, starting at drive C:. If you have a large number of logical drives defined, you may get many screens of information returned to you. To overcome this enter:

DI -h | MORE

This will display the data one page at a time.

DI -p

DI will display the technical information about all of the physical disks attached to your system. An example of this switch is:

Number of physical disks : 1

```
Physical Disk      : 1
  Cylinders        : 1307
  Heads            : 64
  Sectors Per Track : 32
```

DI C: E:

DI will report the information about logical disks C: and E: only.

DI C: -p

DI will report only the physical disk information. No logical disk information about C: is returned.

Networks

A quick note about networks and logical drives is necessary at this point. When you use a network drive, it appears as a logical drive to your system. OS/2 does not support calls of the low level technical nature necessary to return this level of information across networks. If you try DI on a network drive you will get an error message similar to the following:

```
SYS0065: Network access is denied.
```

OS/2 is not inherently based on the FAT file system. It supports other disk formats using installable file systems. On a network, any remote drive to which you are connected may be of a totally different format - one which is not known to OS/2. It may be able to share files, but the internal file structure of the remote file server will not be generally known. For example, the format of UNIX and NETWARE drives could be anything. OS/2 does not need to know the internal formats of network drives. This is why calls of this nature are not supported by OS/2 across networks.

DS - Dir Sort

Summary

DS sorts directory entries on FAT disks in a specified order.

Icon



Command Line Format

Usage: DS {Switches} Sort Order <Start Path> <Drive Spec(s)>

Switches

- a All disks.
- h Hard disks only.
- s Include subdirectories in the search.
- on Run on <Machine Name>.

Note:

If no drive specifications are entered the current drive is the only one displayed.

Sort Order

- D Sort by date order.
- E Sort by Extension.
- N Sort by Name.
- S Sort by Size order.
- T Sort by Time order.
- Reverse the default sort order. The default is to sort from lowest (first) to highest (last). This option reverses this default.

Note:

At least one of the Sort Order options must be entered. Multiple options may be entered, in which case the second and subsequent options are used to further distinguish the sort criteria.

Description

The FAT file system, unlike HPFS, does not sort its directory entries. The files are displayed in the order in which they were found - which can lead to some confusing situations. To prevent this, DS can be used to sort the directory entries.

DS must be able to lock the disk for exclusive access for it to be able to proceed. If it can not, it will inform you that it can not lock the disk and then exit.

If you ask DS to sort a non FAT drive you will receive the following message:

```
Drive is not FAT.
```

Note:

The 16 bit version of DS will only be able to sort approximately 2000 entries per directory. If this number is exceeded, the directory will not be changed. The 32 bit version of DS does not have this restriction.

DS does not currently support a full screen interactive mode.

-a All disks

DS will scan all logical disk drives defined by the system, starting at drive A:. The drives are scanned in order, starting at A:, then B: and so on.

-h Hard disks only

DS will scan all logical disk drives defined by the system, starting at drive C:. The drives are scanned in order, starting at C:, then D: and so on.

-s Subdirectories

DS will include subdirectories as part of its search of the specified disk drives. If a subdirectory name is included as part of the file specification, then the search will include that subdirectory and all subdirectories below it. If no subdirectory is given as part of the file specification, then the current directory and all below it are included in the search.

-on Run on <Machine Name>

This switch forces DEPIPE to be loaded and allows DS to access disk drives located on remote machines. The <Machine Name> must be specified with a leading "\\\". The machine name of the remote machine will have been specified when the network software on the remote machine was installed.

Note:

DESVR must be running on the remote machine for this option to work.

Consult Appendix H - DESVR for further details.

Examples

```
DS en D:\
```

DS will sort all files in the root directory of D:. The files will be sorted by Extension followed by Name.

```
DS -EN G:\ /s
```

DS will sort all of the files on drive G: in reverse order.

```
DS DT D:\WORK
```

DS will sort all of the files in D:\WORK in date and then in time order. If there are multiple files all of the same date, the time will be used to further distinguish them.

```
DS -EN A:\ /s -on \\B5
```

DS will sort all of the files on the A: drive on the machine named B5 in reverse order.

DT - Disk Test

Summary

DT tests the files on a disk for readability. It reads all of the specified files and will report any errors it encounters.

Icon



Command Line Format

Usage: DT {Switches} <File Spec> <Drive Spec(s)>

Switches

- a All disks.
- b Both file and logical disk modes.
- c Cylinder mode.
- d Logical Disk mode.
- h Hard disks only.
- l Keep a Log file.
- p<n> Physical disk mode.
- r ReRun count.
- s Include subdirectories in the search.

Note:

If no drive specifications are entered, only the current drive is tested.

Description

DT tests the files on a disk. It reads all of the specified files and will report any errors it encounters. There are two types of errors which can occur. These are: hard errors, due to hardware failures; and soft errors which are errors that are not due to hardware failures.

Over time, both the media and drive mechanisms can wear. A media error occurs when data written to disk is written to an area which has become

unreliable to read or write. The data which you read may not be the same as that which you wrote. A media error is a hard error, as it is due to a hardware failure. CRC (Cyclic Redundancy Check) and data read errors are examples of hard errors which you may come across.

A soft error is an error which occurs where there is no physical problem with the disk media or drive mechanism. A soft error could be due to data corruption. The data can be read and written correctly with no errors, but the data is not what it should be. For example, in the FAT file system, an entry in the FAT table may be corrupted. It may point to a cluster which does not exist (sector not found error) or it may point to a cluster which is already owned by another file (cross linked clusters). Clearly any given cluster can only belong to a single file. Thus if two (or more) files think that they own the same cluster there is an error.

Some file systems such as HPFS support dynamic hot fixes. If a correctable disk error is detected during a write operation, the data effected by the error is transferred to another location on the disk, and the flawed area marked as such and no longer used. It has been noted that all disk errors are not caught in this fashion. So it may be possible to detect errors which have arisen over time. The FAT file system has no such facility for this type of error detection. Using DT, it is possible to detect if any files are located on a flawed area of a disk.

DT also displays an effective transfer rate of the data which is read from the disk. This is only an approximate value, as it is the transfer rate for the process in which DT is running. Remember - there may be other processes which are accessing the disk at the same time, so your transfer rate could vary considerably.

Note:

DT attempts to open all of the files which it reads in `OPEN_ACCESS_READONLY` and `OPEN_SHARE_DENYNONE` mode. This means that it opens files in read-only mode, and that it does not restrict access to any other processes which may be trying to access the files. Under OS/2, processes can specify the access and share modes of files that they open. Some processes will open their files with a share mode of `OPEN_SHARE_DENYALL`, which means that no other process can open that file. When DT comes across a file which is in use by another process and can not open it, you will see the following error message:

```
SYS0032: The process cannot access the file because
it is being used by another process.
```

and the name of the file which could not be opened will be displayed.

In file test mode, areas of the disk which do not contain files will not be read. Use the -d or -p options to read all areas of the disk.

-a All disks

DT will scan all logical disk drives defined by the system, starting at drive A:. The drives are scanned in order, starting at A:, then B: and so on.

-b Both File and Logical disk mode

The default mode of DT is to only test read files. This switch forces DT to test read all files and all sectors on the logical disk.

-c Cylinder mode

When DT is reading either Logical or Physical disks, it reads one sector at a time. This thoroughly tests the system, however it is very slow. The -c switch, when used in conjunction with either the -d or -p switches, forces DT to read an entire track at a time. Reading an entire track at a time is considerably faster than reading each sector individually.

-d Logical disk mode

This switch causes DT to read all sectors on the specified logical disks. It displays some details about the disk. It also displays number of cylinders (tracks), heads, sectors per track, total sectors and total bytes to be read. The current cylinder, head and sector are displayed as they are being read. DT will report any errors it encounters.

-h Hard disks only

DT will scan all logical disk drives defined by the system, starting at drive C:. The drives are scanned in order, starting at C:, then D: and so on.

-l Keep a Log file

Using this switch causes DT to create a log file named DT.LOG in the current directory. This log file lists what DT does, including any errors it encounters.

-p Physical disk mode

This switch causes DT to read all sectors on the specified physical disk. It displays some details about the disk. It also displays number of cylinders (tracks), heads, sectors per track, total sectors and total bytes to be read. The current cylinder, head and sector are displayed as they are being read. It will report any errors it encounters. Only one physical disk can be read at a time.

-r ReRun count

The -r switch specifies the number of time that an operation is to be performed on a specified area of the disk. For disk mode tests, the sector or complete cylinder will be test read <n> times. For file mode tests, the file will be test read <n> times.

-s Subdirectories

DT will include subdirectories as part of its search of the specified disk drives. If a subdirectory name is included as part of the file specification, then the search will include that subdirectory and all subdirectories below it. If no subdirectory is given as part of the file specification, then the current directory and all below it are included in the search.

Examples

DT -h

DT will attempt to read all files on all hard disks.

DT C:*.EXE

DT will read all .EXE files in the root directory of drive C:.

DT C:*.EXE /s

DT will read all .EXE files in the root directory of drive C: and all of its subdirectories.

DT C:\OS2\DLL\WARPCOMM.DLL

DT will read C:\OS2\DLL\WARPCOMM.DLL only.

DT -b E:

DT will read all of the files on Drive E: and then read all of the sectors on drive E:

DT -p1

DT will only read the sectors on physical disk 1.

DT E: -cd

DT will perform a logical disk test of E:; reading a complete cylinder at a time.

FA - File Attributes

Summary

FA allows you to view and change file attributes.

Icon



Command Line Format

Usage: FA {Switches} <File Spec> <Drive Spec(s)>

Switches

+/-r	Set or clear the Read-Only attribute.
+/-a	Set or clear the Archive attribute.
+/-s	Set or clear the System attribute.
+/-h	Set or clear the Hidden attribute.
/a	All disks.
/f	Force file names to lower case.
/h	Hard disks only.
/s	Include subdirectories in the search.

Note:

To set a file attribute use the +.

To clear a file attribute use the -.

Description

FA allows you to view and change file attributes. If no file specification is given, it is assumed that all files are to be changed. If no changes are requested then the current attributes of the files which match the given file specification are displayed.

For example, if you try to remove a directory (RD) and you get the error message:

```
SYS0016: The directory cannot be removed.
```

then check to see if there are hidden files in that subdirectory. If there are no hidden files, then another process may be using the directory.

Note:

This is the only utility in the Graham Utilities which requires a "/" instead of a "-" character for the other switches.

+/-r Read-Only Attribute

The Read-Only attribute protects files from being modified or deleted. The use of this option protects a file from being accidentally changed or deleted.

+/-a Archive Attribute

The Archive attribute is set by applications to tell the OS/2 BACKUP command (and other backup utilities) that the file requires backing up. These backup utilities will reset (clear) the archive attribute.

+/-s System Attribute

The System attribute indicates that the file is an OS/2 or operating system related file. Files with this attribute do not appear in the standard OS/2 directory listings.

+/-h Hidden Attribute

The Hidden attribute is set to hide files from normal use. Files with this attribute do not appear in the standard OS/2 directory listings. They cannot be deleted, executed and cannot normally be accessed by most programs.

/a All disks

FA will scan all logical disk drives defined by the system, starting at drive A:. The drives are scanned in order, starting at A:, then B: and so on.

/f Force file names to lower case

On FAT partitions, file names are always displayed in lower case. HPFS partitions retain their case, but the file system does not differentiate between the case of file names, and they are displayed as they were entered. The -f option forces all displayed file names, across all file system types, to be displayed in lower case.

/h Hard disks only

FA will scan all logical disk drives defined by the system, starting at drive C:. The drives are scanned in order, starting at C:, then D: and so on.

/s Subdirectories

FA will include subdirectories as part of its search of the specified disk drives. If a subdirectory name is included as part of the file specification, then the search will include that subdirectory and all subdirectories below it. If no subdirectory is given as part of the file specification, then the current directory and all below it are included in the search.

Examples

```
FA -r -a -s -h *.*
```

FA will clear all the attributes of every file in the current directory.

```
FA -r -a -s -h *.* /s
```

FA will clear all the attributes of every file in the current directory, and all subdirectories below it.

```
FA /f *.DLL +r
```

FA will set the file attribute of all .DLL files in the current directory to read only. In addition, the file names will be printed in lower case.

```
FA +r +a +s +h C:\IO.SYS
```

This will set the file attributes of C:\IO.SYS to read only, archive, system and hidden, which is the default state for this file.

FA *.DLL +R /s C: E:

FA will set all .DLL files to read only on the current drive as well as drive C:
and E:.

FA C:*.DLL +R /s

FA will set all DLL's to read only on drive C: only.

FD - File Date

Summary

FD sets all files which match the file specification to the current date and time unless overridden on the command line.

Icon



Command Line Format

Usage: FD {Switches} <File Spec> <Drive Spec(s)>

Switches

-a	All disks.
-d	Specify the date in your local format.
-f	Force file names to lower case.
-h	Hard disks only.
-s	Include subdirectories in the search.
-t	Specify the time in 24 hour format.

Note:

If no file specification is entered, all files are listed. If drive specifications are entered then those drives are searched. If no drive specifications are entered then the current drive is searched.

Description

FD sets all files matching the file specification to the current date and time, unless overridden on the command line. The file's last modification date and time are modified. HPFS partitions support date and time fields for the file's creation and last access. FD does not modify these fields.

-a All disks

FD will scan all logical disk drives defined by the system, starting at drive A:. The drives are scanned in order, starting at A:, then B: and so on.

-d Specify the date

This switch allows you to nominate a specific date instead of using the current date. FD will use the date format of your native country. The available formats are:

DD-MM-YY	Day-Month-Year
MM-DD-YY	Month-Day-Year
YY-MM-DD	Year-Month-Day

-f Force file names to lower case

On FAT partitions, file names are always displayed in lower case. HPFS partitions retain their case, but the file system does not differentiate between the case of file names, and they are displayed as they were entered. The -f option forces all displayed file names, across all file system types, to be displayed in lower case.

-h Hard disks only

FD will scan all logical disk drives defined by the system, starting at drive C:. The drives are scanned in order, starting at C:, then D: and so on.

-s Subdirectories

FD will include subdirectories as part of its search of the specified disk drives. If a subdirectory name is included as part of the file specification, then the search will include that subdirectory and all subdirectories below it. If no subdirectory is given as part of the file specification, then the current directory and all below it are included in the search.

-t Specify the time

Instead of using the current time, this switch allows you to nominate a specific time. The time is specified in 24 hour HH:MM:SS format. 00:00:00 is midnight, 12:00:00 is noon, 23:00:00 is 11pm and so on.

Examples

FD

This will set all files in the current directory to the current date and time.

FD *.exe -d31-12-93

When run in Australia, this will set the date of all .EXE files to the 31st of December 1993.

FD *.exe -d12-31-93

When run in the United States of America, this will set the date of all .EXE files to the 31st of December 1993.

FD *.exe -d93-12-31

When run in Japan, this will set the date of .EXE files to the 31st of December 1993.

FD -hs

This will set the date and time of the last write of all files on all hard disks to the current date and time.

FF - File Find

Summary

FF locates all files matching the given file specification on all specified disk drives.

Icon



Command Line Format

Usage: FF {Switches} <File Spec> <Drive Spec(s)>

Switches

- a All disks.
- d Delete found files.
- f Force file names to lower case.
- h Hard disks only.

Note:

If no file specification is entered, all files are listed. If drive specifications are entered then those drives are searched. If no drive specifications are entered then the current drive is searched.

Description

FF will recursively search your disk drives for specified files. By default, it will start searching from your root directory. Although, if specified, FF can begin the search from a particular subdirectory.

-a All disks

FF will scan all logical disk drives defined by the system, starting at drive A:. The drives are scanned in order, starting at A:, then B: and so on.

-d Delete found files

FF will delete any files found which match the given file specification.

WARNING: You are NOT prompted for confirmation of the deletion of the files.

-f Force file names to lower case

On FAT partitions, file names are always displayed in lower case. HPFS partitions retain their case, but the file system does not differentiate between the case of file names, and they are displayed as they were entered. The `-f` option forces all displayed file names, across all file system types, to be displayed in lower case.

-h Hard disks only

FF will scan all logical disk drives defined by the system, starting at drive C:. The drives are scanned in order, starting at C:, then D: and so on.

Examples

```
FF *.pas c: d: e:
```

Look for any files matching *.pas on drives C:, D: and E:

```
FF *.BAK -d
```

Delete all .BAK files on the current drive.

```
FF *.TMP -hd
```

Delete all .TMP files on all hard disks.

```
FF *.WKS E:
```

Find all .WKS files on the current disk drive as well as drive E:.

```
FF E:*.WKS
```

Find all .WKS files on drive E: only.

FI - File Information

Summary

FI allows you to display and edit comments attached to files.

Icon



Command Line Format

Usage: FI {Switches} <File Spec> <Drive Spec(s)>

Switches

-a	All disks.
-c	Only files with comments attached.
-d	Delete comments.
-e	Edit or add comments interactively.
-h	Hard disks only.
-l	Long comment listing.
-n	Set the comment as needed.
-s	Include subdirectories in the search.

Note:

If no file specification is entered, all files are listed. If drive specifications are entered then those drives are searched. If no drive specifications are entered then the current drive is searched.

Description

FI allows you to attach, view and edit comments to files. FI uses Extended Attributes to save the file comments. See Appendix A for a description of extended attributes.

FI uses the .COMMENTS extended attribute to save its information.

-a All disks

FI will scan all logical disk drives defined by the system, starting at drive A:. The drives are scanned in order, starting at A:, then B: and so on.

-c Only files with comments attached

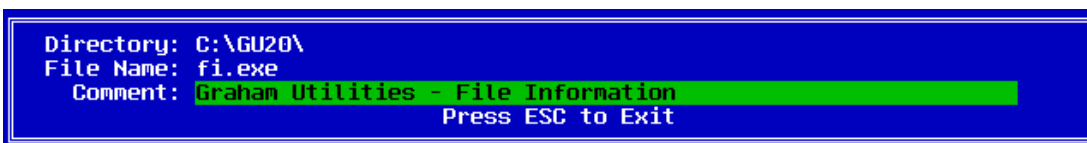
By default, FI lists all files matching the given file specification. This switch forces FI to only display files with comments attached to them.

-d Delete comments

This switch causes FI to delete the comments attached to all files matching the given file specification. No warning or prompting is given with the use of this switch.

-e Edit or Add comments interactively

This switch will cause FI to display a small window at the bottom of the screen and prompt you to edit or add comments for each file which matches the given file specification. The edit window looks like the following:



```
Directory: C:\GU20\  
File Name: fi.exe  
Comment: Graham Utilities - File Information  
Press ESC to Exit
```

-h Hard disks only

FI will scan all logical disk drives defined by the system, starting at drive C:. The drives are scanned in order, starting at C:, then D: and so on.

-l Long comment listing

By default FI lists the file name, size, date, time and comments for each file. This may cause screen wrap, particularly when displaying long file names or long comments. The -l switch causes FI to only display the file name and comments.

-n Set EA as needed

This switch sets the EA as critical or needed.

By default, Extended Attributes (EAs) are non-critical. A non-critical EA is one which is not necessary to the functionality of the application. If a non-critical EA is lost, the system continues to operate correctly. For example, losing the icons associated with data files does not generally cause any ill effect other than the inability to show the icon.

A critical extended attribute is one which is necessary for the correct operation of the operating system or of a particular operation. EAs should be marked as critical if their loss would cause the system or program to perform incorrectly. For example, a mail program might store mail headers in EAs. The loss of the header from a message would normally render the mail program unable to use that message. This would be unacceptable, so the mail program should mark this EA as critical.

-s Subdirectories

FI will include subdirectories as part of its search of the specified disk drives. If a subdirectory name is included as part of the file specification, then the search will include that subdirectory and all subdirectories below it. If no subdirectory is given as part of the file specification, then the current directory and all below it are included in the search.

-w Only files without comments attached

By default, FI lists all files matching the given file specification. This switch forces FI to only display files without comments attached to them.

Examples

FI

FI will list all files. It will display the file name, size, time, date and any comments attached to them.

FI -c

FI will only display the files that have comments attached to them. The file name, size, time, date and the comment will be displayed.

```
FI -l
```

FI will list all files. It will display only the file name and any comments attached to them.

```
FI -cl
```

FI will only display the files which have comments attached to them. It will only display the file name and the comment.

```
FI -d
```

FI will delete all comments attached to all files in the current directory.

```
FI C:\STARTUP.CMD -e
```

FI will prompt you to add or edit comments to C:\STARTUP.CMD only.

```
FI C:\STARTUP.CMD -d
```

FI will only delete any comments attached to C:\STARTUP.CMD.

```
FI -ed
```

FI will delete the comments attached to each file in the current directory and then prompt you to add new comments for each file. This is done on a file by file basis.

```
FI -ew
```

This will enable you to edit all files which currently do not have a comment attached to them.

```
FI -cw
```

This will cause FI to display all files, whether or not they have comments attached.

FS - File Size

Summary

FS lists the file sizes of all files matching the given file specification, on all specified drives.

Icon



Command Line Format

Usage: FS {Switches} <File Spec> <Drive Spec(s)>

Switches

- a All disks.
- f Force file names to lower case.
- h Hard disks only.
- s Include subdirectories in the search.
- t List Totals only, not individual files.

Note:

If no file specification is entered, all files are listed. If drive specifications are entered then those drives are searched. If no drive specifications are entered then the current drive is searched.

Description

FS lists the file sizes of all files matching the given file specification, on all specified drives. It can also list the totals of all files in specified subdirectories.

-a All disks

FS will scan all logical disk drives defined by the system, starting at drive A:. The drives are scanned in order, starting at A:, then B: and so on.

-f Force file names to lower case

On FAT partitions, file names are always displayed in lower case. HPFS partitions retain their case, but the file system does not differentiate between the case of file names, and they are displayed in the same case as which they were entered. The -f option forces all displayed file names, across all file system types, to be displayed in lower case.

-h Hard disks only

FS will scan all logical disk drives defined by the system, starting at drive C:. The drives are scanned in order, starting at C:, then D: and so on.

-s Subdirectories

FS will include subdirectories as part of its search of the specified disk drives. If a subdirectory name is included as part of the file specification, then the search will include that subdirectory and all subdirectories below it. If no subdirectory is given as part of the file specification, then the current directory and all below it are included in the search.

-t Totals Only

By default FS will list the name and size of each file matching the given file specification. This switch will force FS to only display the totals of all files matching the file specification. Also displayed is the slack. This is the difference between the actual file size and the size allocated for the file. It is expressed as a percentage.

Examples

```
FS C:\OS2 -st
```

Display the total size of all files in C:\OS2 and its subdirectories.

```
FS C: E:
```

Lists the names and size of all files in the current directories on drive C: and E:.

FS -st

Lists the names and total size of all files in the current directory and its subdirectories.

FS -ht

Lists the total size of all files in all subdirectories on all hard disks.

GCD - Graham Change Directories

Summary

GCD scans the specified drive and produces a graphic tree representing the structure on the drive. You may move a highlight bar through the listed directories and use the Enter key to change to a particular directory.

Icon



Command Line Format

Usage: GCD {Switches} <Drive Spec\Start Path>

Switches

-i<Path>	Use path to find GCD.INI.
-n	Do NOT save directory data.
-r	Rescan the specified disk.

Note:

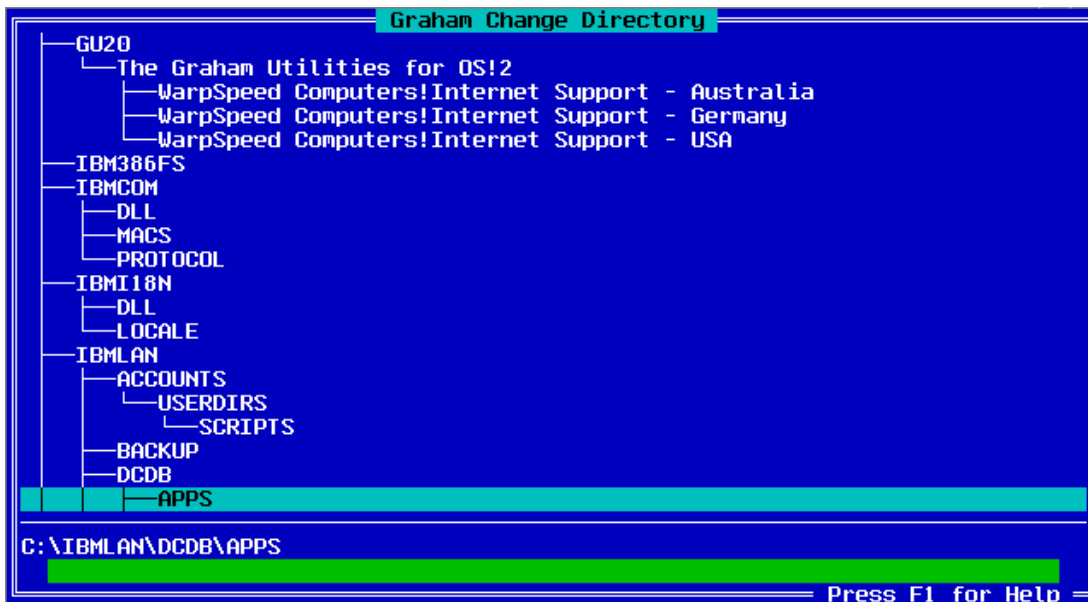
If no drive specifications are given, then the current drive is listed.

Description

GCD scans the specified drive, or if no drive is specified, scans the current drive and produces a graphic tree representing the directory structure of that drive. If an initial path is given, GCD will search for an occurrence of the path in the directory structure. It is not necessary to give the complete directory path component. The specified component is used as a sub string to search within the full directory name, not just the last subdirectory portion of a complete directory path. Thus, if OS2 is specified, GCD will find matches in:

```
\OS2
\OS2UTILS
\MY\OS2
\TEST\_OS2FILES
```

If no start path is given, GCD will highlight the current directory. Whether or not a start path is given, you will be presented with a window similar to the following:

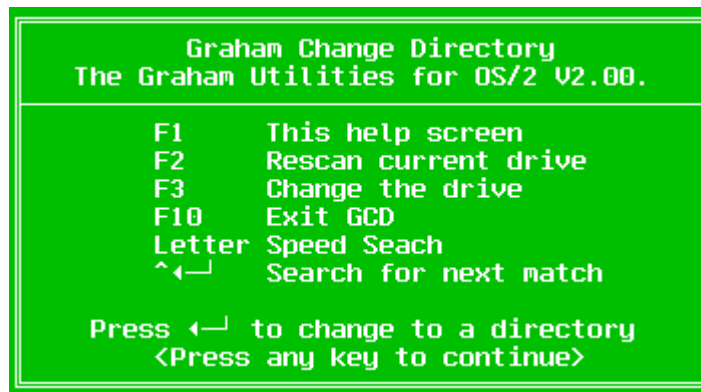


You may use the Up Arrow, Down Arrow, Page Up, Page Down, Control-Home (move to the very top) and the Control-End (move to the very end) keys to change the current directory. The current directory is shown by the highlight bar, as well as being listed at the bottom of the screen.

There is also an edit field at the bottom of the screen. This field is used to enter a partial directory match for a speed search. Every time a key is pressed a partial search is done. Control-Enter can be used to find the next matching entry. Remember, GCD reports a match on any point within the full directory name.

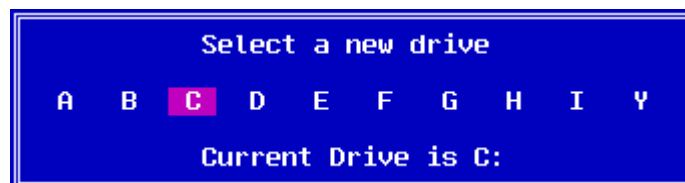
To change to the currently highlighted directory press the Enter key. The program will exit and change to the specified directory. The ESC key will cause the program to exit without changing directories.

Help is available via the F1 key. If pressed you will be presented with this screen:



GCD saves the directory structures of disks which it scans (unless the -n switch was specified) in the GCD.INI file. Each time you select a disk, GCD reads the volume label and uses that information to look up previously saved directory structures. This saves time when loading GCD. However if you have created or removed directories since the last time you ran GCD, the saved data will be incorrect. It is possible to use the F2 key to force a rescan of the drive. GCD will save the newly read data (unless the -n switch was specified), so the next time you run GCD it will have the correct data. Saving data can save a considerable amount of time, particularly if the directory structure on the drive is complex.

If you wish GCD to change drives, press the F3 key and you will be presented with the select drive dialog box, which will look similar to the following:



Use the left or right arrow keys to select the required drive. Press Enter to choose that drive. Alternatively, you can simply press the desired drive letter. Pressing ESC will cause the program to abort and no drive change will occur.

The select drive dialog box can be bypassed by entering a drive specification when GCD is started.

-i Use path to find GCD.INI

By default, GCD uses the GU environment variable to locate the GCD.INI file. Alternatively, you may use this switch to specify an alternative path to locate the GCD.INI file. If GCD.INI is not found, GCD will rescan the directories on the disk. The GU environment variable is set in your CONFIG.SYS file as set by the install program.

-n Do NOT save directory data

By default GCD will save the scanned directories in the GCD.INI file which is pointed to by the GU environment variable. This switch stops GCD from saving the data.

-r Rescan the specified disk

A rescan can be forced by specifying this switch. It will scan the drive and save the data - unless the -n switch was specified.

Examples

GCD

GCD will display the directory structure, as previously saved, on the current drive. If no data has been previously saved, GCD will scan the disk and save the data.

GCD -r

GCD will rescan the current drive, save the structure information, and display the directory tree.

GCD -rn

GCD will display the directory structure on the current drive after rescanning the disk. The data will not be saved.

GCD E:

GCD will display the directory structure on drive E:

GCD E: -r

After having rescanned drive E:, GCD will display the directory structure of drive E:.

GCD E:\OS2UTILS

GCD will display the directory structure on drive E:. The current directory will be set to the first match of "OS2UTILS" (if one exists).

GCD \OS2UTILS

GCD will search for the \OS2UTILS directory on the current drive. If it is not found, the highlight bar will be on the current directory.

GCD -iC:\GU20

GCD will use the file C:\GU20\GCD.INI to save its structure information.

GI - Graham Integrator

Summary

GI is a simple, easy to use guide to The Graham Utilities for OS/2.

Icon



Command Line Format

Usage: GI

Description

GI lists all of the available programs, their uses, and a summary of all of the switches and parameters. It also provides full line editing features so that you may edit the command line which is passed to the programs. Pressing enter will enable the execution of the specified program.

GI will look like this:

```
The Graham Integrator
BE  Batch Enhancer
DI  Disk Information
DS  Dir Sort
DT  Disk Test
FA  File Attributes
FD  File Date
FF  File Find
FI  File Information
FS  File Size
GCD  Graham CD
LD  List Directories
SA  Screen Attributes
SI  System Information
TM  Time Mark
UL  Volume Label
WC  Word Count
2LZH
Beep
Case
Clipbrd
Config

BE - Batch Enhancer
Usage: BE <File Spec>
or
      BE <Command> [parameters]

Where <Command> is one of the following:

ASK      BEEP      BOX      CLS
DAY      DELAY     MONTH    PRINTCHAR
REBOOT   ROWCOL     SA       SEMCLEAR
SEMCLOSE SEMCREATE  SEMKILL  SEMLIST
SEMSET   SEMWAIT    WEEKDAY  WINDOW
YEAR

Enter BE <Command> ? to get help on a specific
command.
BE can be used to enhance batch files (.CMD).
The semaphore functions can be used to synchronise
tasks running in different sessions.

BE
```

Press F1 for Help

The F1 key will show you this help screen:

```

                                The Graham Integrator
                                The Graham Utilities for OS/2 V2.00.
                                Copyright (C), Chris Graham - WarpSpeed Computers 1994-1996
                                Designed and Written by Chris Graham.

You can execute any of the Graham Utilities from within GI.
Move the highlight to the command, type any arguments, and
press Enter.

F2                View complete online help.
Enter             Run the program on the command line.
↑ or ↓          Move to previous or next command.
Letters          Type arguments on the command line.
Esc, F10        Quit GI.

                <Press any key to continue. F1 for more.>
```

If you press F1 for further help, you will be presented with my contact details:

```

The author of these utilities can be contacted at:

                                Chris Graham
                                WarpSpeed Computers
                                PO Box 212
                                Brunswick VIC 3056
                                Australia

                                Phone                EMail
                                -----                -----
Voice:  +61-3-9384-1060  FidoNet:    3:632/344
FAX:    +61-3-9386-9979  Internet:   100250.1645@CompuServe.Com
BBS:    +61-3-9386-3104  CompuServe: 100250,1645
                                N,8,1 ANSI    Web Pages:
                                300-28,800    http://www.netins.net/showcase/spectre
                                http://www.warpspeed.com.au
```

Example

GI

There are no options for GI.

LD - List Directories

Summary

LD will list the directories on specified disks.

Icon



Command Line Format

Usage: LD {Switches} <Drive Spec(s)>

Switches

- a All disks.
- g Produce a Graphic directory tree.
- h Hard disks only.

Note:

If no drive specifications are given, then the current drive is listed. The default output is the directory path names.

Description

LD produces a directory listing of your disks. It can produce either a list of directory names, or a graphic directory tree listing.

-a All disks

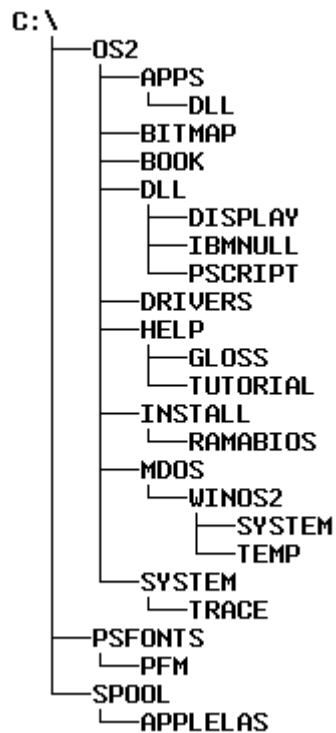
LD will scan all logical disk drives defined by the system, starting at drive A:. The drives are scanned in order, starting at A:, then B: and so on.

-g Produce a Graphic directory tree

The default mode of LD is to display the list of directories as they would normally be printed. For example:

```
C:\
C:\OS2
C:\OS2\APPS
C:\OS2\APPS\DLL
C:\OS2\BITMAP
C:\OS2\BOOK
C:\OS2\DLL
C:\OS2\DLL\DISPLAY
C:\OS2\DLL\IBMNULL
C:\OS2\DLL\PSCRIPT
C:\OS2\DRIVERS
C:\OS2\HELP
C:\OS2\HELP\GLOSS
C:\OS2\HELP\TUTORIAL
C:\OS2\INSTALL
C:\OS2\INSTALL\RAMABIOS
C:\OS2\MDOS
C:\OS2\MDOS\WINOS2
C:\OS2\MDOS\WINOS2\SYSTEM
C:\OS2\MDOS\WINOS2\TEMP
C:\OS2\SYSTEM
C:\OS2\SYSTEM\TRACE
C:\PSFONTS
C:\PSFONTS\PFM
C:\SPOOL
C:\SPOOL\APPLELAS
```

The -g switch forces LD to produce a graphic directory tree listing similar to:



The directory tree gives a more intuitive feel as to how the directories are nested.

-h Hard disks only

LD will scan all logical disk drives defined by the system, starting at drive C:. The drives are scanned in order, starting at C:, then D: and so on.

Examples

LD

This will produce a listing of subdirectories on the current drive.

LD /G

This will produce a graphic directory tree for the current drive.

LD C:

LD will produce a listing of the subdirectories on drive C:.

LD /G C: D: E:

Will produce a graphic directory tree for drives C:, D: and E:.

LD -gh

This will produce a graphic directory tree listing for all hard disks.

SA - Screen Attributes

Summary

SA allows the user to specify the current screen attributes, or colours.

Icon



Command Line Format

```
Usage: SA [bright] [blinking] [reverse] [underline]
        <colour> <on colour>
```

Parameters

- | | |
|------------------|--|
| bright | Specifies that the foreground colour will be bright. This is an optional parameter. |
| blinking | Specifies that the displayed characters will blink or flash. This is an optional parameter. |
| reverse | Specifies that the text to be displayed will be displayed in black on white, unless over ridden by a foreground/background combination. This is an optional parameter. |
| underline | Specifies that the text to be displayed will be underlined. This will only work on monochrome monitors. Colour monitors will display any underlined text in blue. |
| colour | Specifies the foreground colour. It can be any one of the valid colours. This is an optional parameter. |
| on colour | Specifies the background colour. It can be any one of the valid colours. This is an optional parameter. |

The valid colours are:

Black
Red
Green
Yellow
Blue
Magenta
Cyan
White

Description

SA allows the user to specify the current screen attributes, or colours. It sets the current screen attributes so that any further displayed text will appear in those attributes unless overridden with a colour specification from another SA or similar command.

Examples

```
SA RED
```

This will set the current foreground colour to red. The background colour is set to black.

```
SA ON RED
```

This will set the current background colour to white on red. If the foreground colour is not specified, it will default to white.

```
SA Bright green
```

This will set the foreground colour to bright green on a black background.

```
SA REVERSE
```

This will set the current text colours to black text on a white background.

```
SA blinking red
```

This will set the current foreground text colour to blinking red on a black background.

SI - System Information

Summary

System Information displays technical information regarding your hardware and operating system.

Icon



Command Line Format

Usage: SI

Description

SI displays some of the information which OS/2 maintains on your system. The items which it returns are:

Machine Type	This field reports on the type of the machine. If possible, it will list the class and type of the machine.
Operating System	This field reports on the Operating System version which is currently running.
OS/2 boots from	The drive on which OS/2 was installed and started from.
Country Code	This is the country code setting for the current session.
Country Code Page	This is the current country code page (or character set) setting for the current session.
Maximum sessions	The maximum number of sessions available, for both Presentation Manager and VIO or Full Screen sessions.

Maximum VIO sessions	The maximum number of VIO or Full Screen sessions.
Maximum PM sessions	The maximum number of Presentation Manager sessions.
Max Wait	This field specifies the maximum number of seconds which a process has before the OS/2 scheduler devotes CPU time to another process. This parameter is configured in the CONFIG.SYS file.
Timeslice	<p>This field specifies the Maximum and Minimum time slice of the system. The priority is also listed - it is either Absolute or Dynamic.</p> <p>The timeslice parameters dictate the length of time which the OS/2 scheduler uses. If a thread uses up its entire timeslice, it must be pre-empted. The next timeslice for that thread is one tick longer, up to the specified maximum.</p> <p>If the priority is dynamic (the system default), the priority of threads can be adjusted by the system, to ensure that all processes receive some CPU. If the priority is absolute, thread priorities can not be adjusted, and the MaxWait parameter has no effect.</p>
Timer Interval	This field details the system timer interval. It is specified in micro seconds (uSec).
Current Video Mode	<p>This shows the details of the current video mode. It will return information in the form of:</p> <p><Text/Graphics>, <Columns> x <Rows> <Colour/Monochrome></p>
Co-Processor	This field reports on whether or not you have a maths co-processor installed. Remember that OS/2 2.x will emulate one if necessary. Therefore SI will report one even if you do not have one installed.
Parallel Ports	The number of parallel ports available to the system.
Serial Ports	The number of serial ports available to the system.

System up since The last time that the system was booted. An example is:
Fri Jan 07 21:00:15 1994

Note:
This time will wrap around after approximately 45 days.

Valid Drives A list of the valid disk drives available to the system - both local and remote.

Current Adaptor This field displays a comment on the current video display adaptor. The predefined values are:

Monochrome/printer adaptor
Colour graphics adaptor
Enhanced graphics adaptor
Video graphics array display adaptor
IBM Personal System/2 display adaptor 8514/A
Image Adaptor
XGA graphics adaptor
Super video graphics array display adaptor
Unknown display adaptor (unknown code)

The unknown code will be displayed as a number. This indicates that the value is one reserved for future adaptors.

Current Display This field displays a comment on the current video display. The predefined values are:

- Monochrome display
- Colour display
- Enhanced colour display
- 8503 monochrome display
- 8512 or 8513 colour display
- 8514 colour display
- IBM plasma display
- Monochrome 8507 or 8604
- 8515 display
- 9515 display
- 9517 display
- 9518 display
- OEM display
- Unknown display type (unknown code)

The unknown code will be displayed as a number. This indicates that the value is one reserved for future adaptors.

Display Adaptor Memory The size of the video memory on the current display adaptor in bytes.

ANSI processing This reports on whether ANSI escape sequence processing is on or off for the current session.

Example

SI

There are no switches or parameters for SI.

TM - Time Mark

Summary

TM displays the current time and date in a more readable format than OS/2 provides. It also provides multiple timer support.

Icon



Command Line Format

Usage: TM { [Commands] <Switches> }

Switches

-c<n>	Specify timer <n>.
-l	Write on the left side of screen.
-n	No current time or date display.

Note:

The default output is the time and date on the right side of the screen.

Commands

start	Start the timer specified with the -c switch.
stop	Stop the timer specified with the -c switch.

Description

TM displays the current time and date (in a more readable format than OS/2 provides), via the DATE and TIME commands. It also provides multiple timer support. You can use these timers as stop watches. Because the information is stored in your OS2.INI file, the timers will work even after your machine has been restarted.

-c Start or Stop timer

When the Start or Stop parameter is given, this switch starts or stops the given timer number respectively. Valid timers are from 1 to 32767.

-l Write on the left side of screen

By default, TM displays the date and time information on the right hand side of the screen. The -l switch forces TM to display the date and time on the left hand side of the screen.

-n No current time or date display

This switch tells TM not to display the current date or time. Its main use is in conjunction with the timer displays.

Examples

```
TM start -c99 -n
```

Starts timer 99. No current time and date is displayed.

```
TM start -c1 -c2 -c3
```

This starts timers 1, 2 and 3. The current time and date is displayed.

```
TM stop -c2 -c3
```

This displays the elapsed time for timers 2 and 3. It does not actually stop the timer, as the original start time is saved in the OS2.INI file. What TM does is to calculate the time difference and display the result. This gives you the ability to continually stop and monitor the progress of a timer - similar to "lap" timing.

VL - Volume Label

Summary

VL allows you to create, edit and delete the volume labels of disks.

Icon



Command Line Format

Usage: VL <Drive Spec> <"Volume Label">

Note:

If the drive and/or volume label are not given on the command line, you will be prompted for them.

Description

VL allows you to create, edit and delete the volume labels of disks. Although some file systems such as HPFS allow long file names of up to 254 characters, you are still restricted to eleven character upper case volume labels. You may enter a volume label in lower case - however, OS/2 will convert it upper case.

Pressing Enter will attempt to set the volume label to the text in the edit field. Press Escape to leave the volume label as is. The Delete key will remove the current volume label. Totally blank volume labels (all spaces) are not allowed by OS/2. In this event, OS/2 will delete the current volume label.

Examples

```
VL
```

This will display the current volume label of the current disk and prompt you to enter a new one.

VL D:

This will display the current volume label of drive D: and prompt you to enter a new one.

VL C: "System Disk"

VL will set the volume label of drive C: to "SYSTEM DISK".

WC - Word Count

Summary

Word Count is a simple utility which enables you to count the number of characters, words and lines in text files.

Icon



Command Line Format

Usage: WC {Switches} <File Spec> <Drive Spec(s)>

Switches

- a All disks.
- h Hard disks only.
- s Include subdirectories in the search.

Note:

If no file specification is entered, all files are listed. If drive specifications are entered then those drives are searched. If no drive specifications are entered then the current drive is searched.

Description

Word Count is a simple utility which enables you to count the number of characters, words and lines in text files. It lists the individual characters, words and lines of each file as well as the totals of each.

-a All disks

WC will scan all logical disk drives defined by the system, starting at drive A:. The drives are scanned in order, starting at A:, then B: and so on.

-h Hard disks only

WC will scan all logical disk drives defined by the system, starting at drive C:. The drives are scanned in order, starting at C:, then D: and so on.

-s Subdirectories

WC will include subdirectories as part of its search of the specified disk drives. If a subdirectory name is included as part of the file specification, then the search will include that subdirectory and all subdirectories below it. If no subdirectory is given as part of the file specification, then the current directory and all subdirectories below it are included in the search.

Examples

```
WC *.c
```

WC will look for all .C files and list the number of lines, words and characters in each, as well as the total number of lines, words and characters.

```
WC C:\*.TXT /s
```

WC will display the number of characters, words and lines of all .TXT files on drive C:.

2LZH - Automatic archive conversion

Summary

2LZH converts .ZOO, .ZIP, .ARC and .ARJ files to .LZH files. Unless otherwise specified, it automatically searches for files on the current drive.

Icon



Command Line Format

Usage: 2LZH {Switches} <Drive Spec(s)> <File Spec>

Switches

-a	All disks.
-c	Compact mode; keeps the smaller of the compressed files.
-f	Modify Fidonet style FILES.BBS files.
-h	Hard disks only.
-i<Path>	Use path to find 2LZH.INI.
-l	Keep a Log file.
-n	No recursion.
-o	Old (-lh1-) compatibility mode.
-p	Preserve original archive time and date.
-r	Repack *.LZH files.
-s	List file Statistics.
-u	Update mode; replace duplicates.
-v	Virus scan unpacked files.

Description

2LZH automatically scans the specified disk and converts all .ARJ, .ZIP and .ZOO archives found into .LZH format. Peter Fitzsimmon's LH program is an OS/2 version of the DOS LHA and LHARC programs. It fully supports HPFS long file names and extended attributes.

If told to, 2LZH will convert a specific file. The default mode is to convert all archives.

2LZH can also convert nested archives, as well as archives with paths stored in them.

For 2LZH to successfully convert the archives which it encounters, there needs to be sufficient disk space to hold the files in the archive in their unpacked state.

-a All disks

2LZH will scan all logical disk drives defined by the system, starting at drive A:. The drives are scanned in order, starting at A:, then B: and so on.

-c Compact mode; keeps the smaller of the compressed files

In this mode, 2LZH will look at the file sizes of the original archive as well as the new LZH archive. It will keep the smaller of the two.

-f Modify Fidonet style FILES.BBS files

This switch is primarily of interest to BBS operators. It looks for a file "FILES.BBS" in the directory which the original archive was in. If the file is successfully converted, 2LZH will convert the old file name to the new file name in FILES.BBS - if it exists.

-h Hard disks only

2LZH will scan all logical disk drives defined by the system, starting at drive C:. The drives are scanned in order, starting at C:, then D: and so on.

-i Use path to find 2LZH.INI

By default 2LZH uses the GU environment variable to locate the 2LZH.INI file. Alternatively you may use this switch to specify a different path to locate the 2LZH.INI file. If 2LZH.INI is not found, 2LZH will attempt to continue with default values. The default values are the same as those specified in the 2LZH.INI file. The 2LZH.INI file is provided so that you may configure the operation of 2LZH. See Appendix C for complete details.

-l Keep a Log file

2LZH will keep a log file of all important events, as well as the date and time at which they occurred. If the statistics switch is given, they are also recorded in the log file. The log file, 2LZH.LOG, is saved in the directory from which 2LZH is started.

-n No recursion

2LZH will NOT recursively search your disk directory structure. This is so that you can convert only one particular subdirectory.

-o Old (-lh1-) compatibility mode

By default LH uses the newer -lh5- format which gives better compression than the original -lh1- format. Use this switch to force LH to revert to the older format.

-p Preserve original archive time and date

By default, 2LZH will create an archive with the current date and time. This switch sets the new archive to the original archive's date and time.

-r Repack *.LZH files

By default 2LZH will not convert the already existing .LZH archives. Use this switch if you wish to repack the existing LZH files. It is most useful when converting archives between -lh1- and -lh5- formats. To convert new format (-lh5-) archives to the old (-lh1-) format, combine the -r and -o switches together. To update old format archives to the new format, only specify the -r switch.

-s List file Statistics

This switch will force 2LZH to display the file statistics of the converted archives. It lists the number of archives converted and the total size reduction from the old to new archive sizes. If the -l switch is given, the statistics are also recorded in the log file.

-u Update mode; replace duplicates

By default, if 2LZH attempts to convert an archive and there is already a .LZH file of the same name there, the conversion of that archive will be aborted. This update mode switch forces 2LZH to delete the original .LZH file (if present) first, and then convert the other archive.

-v Virus scan unpacked files

This switch forces 2LZH to virus scan the files when they are in their unpacked state. It will report any errors and if any viruses were found abort the archive conversion. See Appendix C for further details on how to configure the operation of the virus scanning software.

Examples

```
2LZH
```

This is the simplest way in which 2LZH can be called. It will convert all archives found on the current drive to LZH format.

```
2LZH -flspu
```

This combination of switches causes 2LZH to modify a FILES.BBS file if one is found, to keep a log file, to report on (and log) statistics, and to preserve the original archive's time and date. In addition, any duplicate files will be replaced with the newly repacked archive.

```
2LZH -flsp H:\Max\File\S_Upload\*.ZIP
```

This will cause 2LZH to convert all ZIP files in H:\Max\File\S_Upload to LZH format. In addition, FILES.BBS is modified, a log file is created, file statistics are noted and the original date and time are kept.

```
2LZH -flsph
```

This will convert all archives found on all hard disks. 2LZH will also modify a FILES.BBS file if one is found, keep a log file, report on (and log) statistics, and preserve the original archive's time and date.

2LZH -rv

2LZH will repack all LZH files on the current drive into the newer, smaller -lh5-format. At the same time it will also convert all of the other archive formats. When unpacked, the files within the archives will be scanned for viruses.

2LZH -ro

2LZH will repack all LZH files on the current drive into the older, larger -lh1-format. At the same time it will also convert all of the other archive formats to -lh1- format.

2LZH -flspiC:\GU20

2LZH will modify a FILES.BBS file if one is found, keep a log file, report on (and log) statistics, and preserve the original archive's time and date. The 2LZH.INI file in directory C:\GU20 will be used.

FILES.BBS files

The FILES.BBS file is used by some FidoNet style BBS programs to store the names and comments of the files which are available to BBS users. Maximus and Opus are examples of BBS programs which use the FILES.BBS format. Other BBS programs may use different formats to store their file lists.

The FILES.BBS file simply contains the file name followed by the comments for that file.

A sample of a FILES.BBS before 2LZH has processed it may look like this:

```
- The archived archive programs themselves
PKZ102-2.ZIP ( 1)OS/2 version of PKZip 1.02
ZOO21E.ZIP ( 2)OS/2 version of ZOO.EXE (2.1) and source
UNZ50X32.ZIP ( 35)OS/2 32 bit UnZip supports new deflation
ZIP19X32.ZIP ( 40)OS/2 32 bit Zip supports new deflation
UNARJ.ZIP ( 1)OS/2 Demo version of UnARJ
GZIPTAR.ZIP ( )OS/2 GZIP and TAR executables (and DLLs)
```

The same file after 2LZH has modified it would look like this:

```
- The archived archive programs themselves
PKZ102-2.LZH ( 1)OS/2 version of PKZip 1.02
ZOO21E.LZH ( 2)OS/2 version of ZOO.EXE (2.1) and source
UNZ50X32.LZH ( 35)OS/2 32 bit UnZip supports new deflation
ZIP19X32.LZH ( 40)OS/2 32 bit Zip supports new deflation
UNARJ.LZH ( 1)OS/2 Demo version of UnARJ
GZIPTAR.LZH ( )OS/2 GZIP and TAR executables (and DLLs)
```

BEEP - Plays tunes

Summary

BEEP plays a tune file.

Icon



Command Line Format

Usage: BEEP <file>

Options

The following options are valid in tune files:

/Fn	Specify the frequency of the tone in Hertz.
/Dn	Specify the duration of the tone in n/18 seconds.
/Tn	Specify the duration of the tone in milliseconds.
/Rn	Repeat a tone n times.
/Wn	Wait n/18 seconds between tones.

They can also be specified in raw Frequency Duration pairs. This makes BEEP compatible with Maximus BBS format tune files.

The repeat and wait options only apply to the line on which they appear.

Tune files may also have comments in them. Comments start with a semicolon ";".

Description

BEEP allows you to play tunes on your computer. It supports a wide variety of input formats. BEEP is capable of parsing the Frequency, Duration pairs of each tone from a variety of text formats. BEEP is not capable of playing WAV or other digitised sound format files.

Note:

BEEP may conflict with 4OS2's internal command BEEP. In this instance the BE BEEP command may be used, or BEEP may be renamed to something else.

Two sample tune files have been provided. They are INDIANA and SIMPSONS.

Examples

```
BEEP INDIANA
```

Beep will play the "Indiana Jones" theme.

```
BEEP SIMPSONS
```

This will cause beep to play the theme from "The Simpsons".

Case - Change the case of files

Summary

Case allows you easily change the case of both files and directories.

Icon



Command Line Format

Usage: Case {Switches} <File Spec> <Drive Spec(s)>

Switches

-a	All disks.
-d	Directories are included.
-f	First letter of each filename segment is upper case, rest are lower case.
-h	Hard disks only.
-l	Lower case file names.
-s	Subdirectories are included.
-u	Upper case file names.
-w	first letter of each Word is upper case, rest are lower case.

Note:

If no drive specifications are entered the current drive is the only one displayed. As well, one of the -f, -l, -u or -w switches must be entered.

Description

Case allows you to easily change the case of the names of both files and directories on your disks. File systems such as HPFS retain the case, but are case insensitive. Thus changing the case of a file does not change the actual file referenced, ie. 'a' is the same as 'A'. UNIX type file systems are case sensitive, so the file 'a' is a totally different file to 'A'. Case should not be run on case sensitive file systems. Under the FAT file system, which is case insensitive, the filenames are always in upper case, so running Case on a FAT disk is useless.

-a All disks

Case will scan all logical disk drives defined by the system, starting at drive A:. The drives are scanned in order, starting at A:, then B: and so on.

-d Directories

Subdirectory names will be changed along with the files in them.

-f First letter of each filename segment is upper

The first letter of each filename segment, as separated by a period ('.') will be changed to upper case. The rest of the letters will be changed to lower case.

-h Hard disks only

Case will scan all logical disk drives defined by the system, starting at drive C:. The drives are scanned in order, starting at C:, then D: and so on.

-l Lower case file names

Every letter of the filename will be changed to lower case.

-s Subdirectories

Case will include subdirectories as part of its search of the specified disk drives. If a subdirectory name is included as part of the file specification, then the search will include that subdirectory and all subdirectories below it. If no subdirectory is given as part of the file specification, then the current directory and all below it are included in the search.

Note:

The directories themselves will not be changed with this switch - only the files in the directories. See also the -d switch.

-u Upper case file names

Every letter of the filename will be changed to upper case.

-w first letter of each Word is upper

This switch is similar to the -f switch. However, each filename segment is now separated by a period or a space.

Examples

Case `C:\ /s -l`

All files on C: will be changed to lower case.

Case `C:\ /s /u`

Case will change all files and directories to upper case.

Case `"H:\this is a very long file.name" -f`

Case will change "H:\this is a very long file.name" to "H:\This is a very long file.Name"

Case `"H:\this is a very long file.name" -w`

Case will change "H:\this is a very long file.name" to "H:\This Is A Very Long File.Name"

Clipbrd - Copy text to the clipboard

Summary

Clipbrd is a fast, simple way of copying text into the clipboard from a command line session.

Icon



Command Line Format

Usage: Clipbrd {Switches} <File Spec>

Switches

-l Large clipboard (>64K)

Description

Clipbrd is a program which allows you to copy text into the system clipboard. By default, Clipbrd takes a file specification as a parameter, although, it can also be used as a pipe. Files or the output of other programs can be piped or redirected into Clipbrd. Presentation Manager programs can then paste that information elsewhere.

-l Large Clipboard (>64K)

Clipbrd will use a large clipboard. Some programs are capable of working with very large amounts of text which may exceed 64K.

Note:

This switch is only effective on the 32 bit version of Clipbrd. The 16 bit version of Clipbrd will report an error and only use the first 64K of data.

Examples

```
Clipbrd C:\Config.sys
```

Clipbrd will copy C:\CONFIG.SYS into the clipboard.

```
DIR | Clipbrd
```

The output of the DIR command will be piped through Clipbrd into the system clipboard.

```
Clipbrd < C:\Autoexec.bat
```

The file C:\AUTOEXEC.BAT will be redirected as input into Clipbrd. The file is copied to the system clipboard.

Config - Configure System Settings

Summary

Config allows you to configure the various configuration options for The Graham Utilities for OS/2.

Icon



Command Line Format

Usage: Config {Switches}

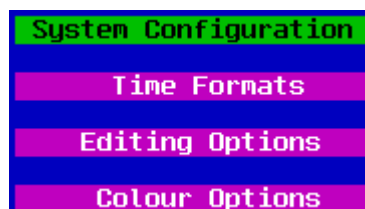
Switches

-r Reset to default values.

Description

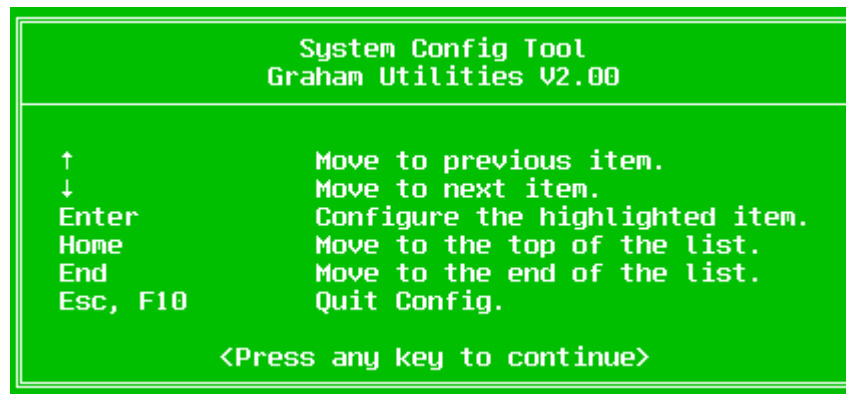
Config allows you to change all of the configurable options of The Graham Utilities for OS/2. A simple menu structure is provided and a short line of context sensitive help is provided at the bottom left hand corner of the window.

The initial menu options for Config are:



Where appropriate, a help panel is provided. It is accessed by pressing the F1 key. Help is available when the 'Press F1 for Help' message is displayed in the lower right hand corner of the window.

A sample help dialog box is presented here:



The main menu gives you four options. They, and their sub-menus are described in detail below:

System Configuration

The only choice available under the System Configuration menu is the DE Popup Dialog option.



The DE Dialog Popup option has two configuration options. They are TRUE or FALSE.

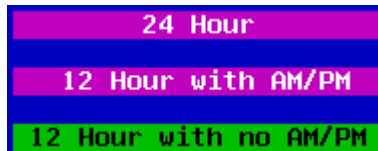
True, which indicates that the dialog box popup will be displayed when any of the DE*.DLL's are loaded by a program, and False, which indicates that the dialog box popup will not be displayed. The dialog box is either displayed or it is not depending on the current setting.



The default option is TRUE, indicating that the dialog box popup will be displayed.

Time Formats

The Time Formats menu has three options.



As you select each one, the current time and date are displayed in the appropriate format.

This is for the first option, 24 hour or ISO 8601 format:



This is for the second option, 12 hour with an AM/PM indicator:



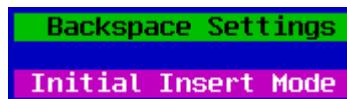
This is for the third option, 12 hour without any AM/PM indicator.



The system default is the last option, 12 hour without any AM/PM indicator.

Editing Options

The editing options menu has two choices.

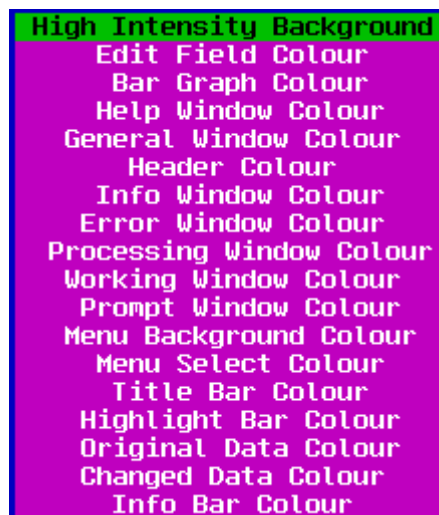


The first option, Backspace Settings, allows you to set whether the backspace key is destructive or not. A non destructive backspace acts as a left arrow, no characters are deleted. The default for this option is FALSE, indicating that the backspace key is not destructive.

The second option, Initial Insert Mode allows you to set the initial insert state of the editing fields. Edit fields can have insert mode initially set on TRUE (on) or FALSE (off). The default for this option is FALSE - insert mode is initially off.

Colour Options

Almost all of the colours used through out all of the programs are user definable. The Colour Options allows you to set these colours and colour options.



As each option is chosen, a sample may be provided to show you the effect of the current values.

High Intensity Background

The high intensity background attribute effects the upper 128 attributes of the 256 available attributes. If this is set the upper 128 attributes have a bright background. If this is not set the upper 128 attributes are interpreted as blinking attributes. The system default is FALSE - the upper 128 attributes are blinking attributes.

Note:

In a VIO Window all blinking attributes will be displayed as high intensity. You may need to use a Full Screen session to see the full effect.

Edit Field Colour

The Edit Field colour is the colour of the edit string fields. You should choose a good high contrast colour combination which you find easy to read.

When editing this colour you will be shown this sample:

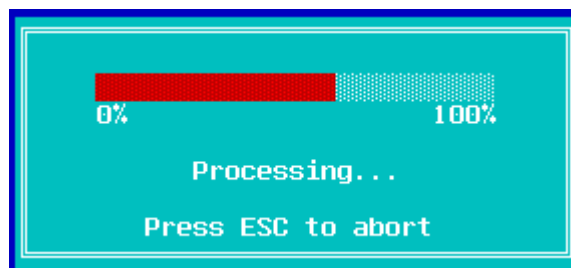


This is a sample Edit Field

Bar Graph Colour

This option allows you to edit the colour of the bar graph as used in text windowed applications. This colour will not effect applications such as SPACE.

This is the sample which is displayed to you whilst editing this option:



Help Window Colour

This is the colour of the help windows which are displayed to you when you press F1.

General Window Colour

This is the colour of the general window backgrounds. Most of the main windows use this colour.

Info Window Colour

This is the colour of any window which is used to display some information or status to you.

Error Window Colour

This is the colour used to specifically display errors to you.

Processing Window Colour

This is the colour of a processing window. A processing window is defined as a process which the application is doing that can not be interrupted. They are generally of very short duration.

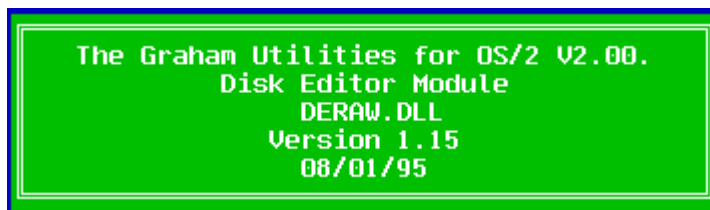
Working Window Colour

This is the colour of a working window. A working window is similar to the processing window, except that it can be interrupted.

Prompt Window Colour

This is the colour of a prompt window. A prompt window is a window where you are being prompted to enter some information which the program requires.

The Help Window, General Window, Info Window, Error Window, Processing Window, Working Window and Prompt Window options all use this as a sample to work upon:



Header Colour

This is the colour of various system headers, typically used for editing. They are generally displayed in the top line of windows. The DE*.DLL's and HexEdit use this option.

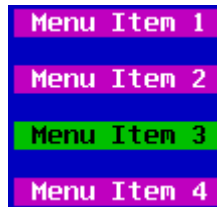
Menu Background Colour

This is the colour of a non selected item on a menu.

Menu Select Colour

This is the colour of the currently selected menu item.

The Menu Background Colour and Menu Select Colour both use this as a sample:



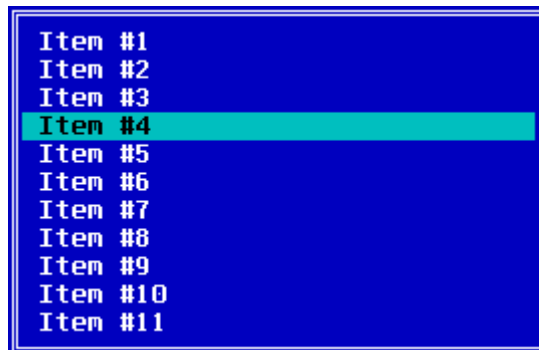
Title Bar Colour

This option allows you to edit the colour of the title bar. This sample is used to guide you:



Highlight Bar Colour

This is the colour of the highlight bar which is used to select/navigate your way through page fulls of information. This is the sample which is used:



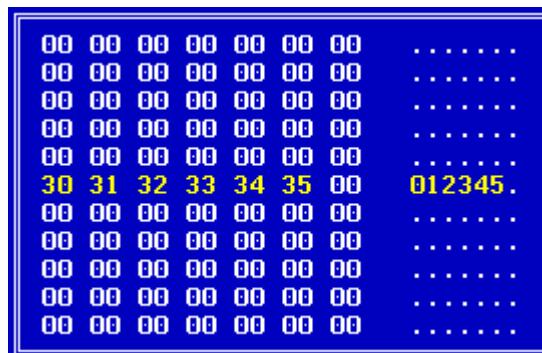
Original Data Colour

This is the colour of the original data. It should have the same background colour as the General Window colour.

Changed Data Colour

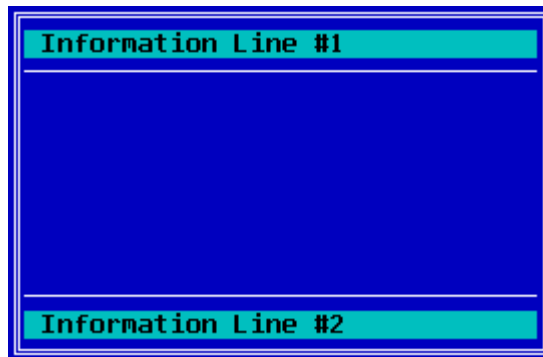
This is the colour of any changed data. It should have the same background colour as the General Window colour.

Both the Original Data and Changed Data colour options use this as a sample to guide you:



Info Bar Colour

This is the colour of the information bars which are used by the FAT-View and HPFSView programs. This is the sample for this option:



-r Reset to default values

Config will reset all settings to the system default settings.

Examples

Config

Start config and edit the existing values.

Config -r

Start config after having reset all of the system values to the initial default. Configuration will then proceed from that point onwards.

CSDInst - Install CSD Files

Summary

CSDInst is a program which allows you to install CSD files.

Icon



Command Line Format

Usage: CSDInst <CSD File>

Description

CSDInst allows you to install CSD file for The Graham Utilities for OS/2. A CSD is IBM speak for Customer Serviceable Diskette. In other words, it is an update disk. Updates, enhancements, additions and bug fixes will be distributed electronically. They will be available from many of the normal electronic sources (CompuServe and the Internet) as well as WarpSpeed Computer's own updates server. Please consult Appendix J for a complete description of WarpSpeed Computer's Internet based services.

CSDInst must be run from the directory in which the utilities were installed.

CSDInst will install new programs or updates to existing programs. For it to be able to do this, any of the programs which form The Graham Utilities for OS/2 must be stopped. OS/2 locks any programs or DLL's which are being run for exclusive use. Thus CSDInst will not be able to update them. CSDInst will automatically bind Extended Attributes to files. It also has the capability to run scripts which are contained within the CSD File as necessary. The SYSLEVEL.GRA file is also updated to reflect the changes.

Once the CSD File has been successfully unpacked and applied the updated NEW-INFO.TXT file will be displayed. This file contains a description of all of the changes and updates contained within the CSD File.

CSD files are cumulative, that is, the most recent CSD contains all of the fixes from previous CSD's as well as the current one.

The CSD files are named in a particular way. The file name describes the product type, family type as well as the date of release for that CSD.

The CSD file name is encoded as such.

GUFFYDDD.CSD

Where the components of the name are described below:

GU This is the product type. GU indicates The Graham Utilities for OS/2.

FF This is the Family type. The current types are:
16 16 bit OS/2.
32 32 bit OS/2.
PP OS/2 for the Power PC.

Y This is the year of release. 1996 would be indicated as 6.

DDD This is Julian day of release. This is the day number of the year so far.

Example

CSDInst GU166205.CSD

CSDInst will install a CSD for The Graham Utilities for OS/2, 16 bit version. The CSD was released on the 205th day of 1996 (the 24th of July).

DIAG - System Diagnostics

Summary

DIAG allows you to report on and test various features of your PC.

Icon

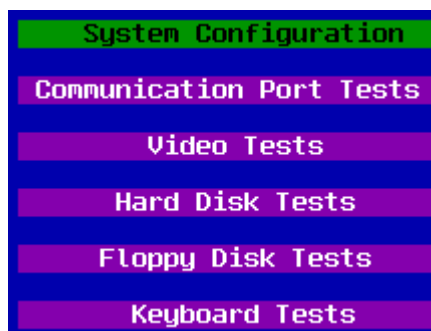


Command Line Format

Usage: DIAG

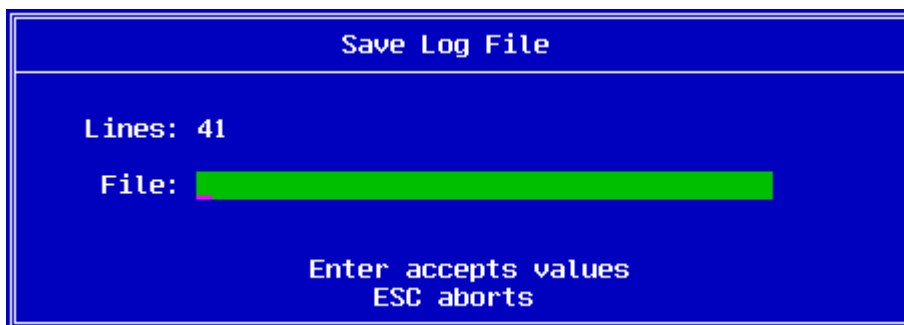
Description

DIAG is a diagnostic tool to test and report on features of your OS/2 system. It uses a simple, nested menu system to navigate through the tests you wish to perform. When DIAG is first started you will see the following menu centred in the middle of the screen:



From here you may select the particular test which you wish to run.

As each test is being run the information is recorded in memory. When you exit the DIAG program (from the main menu), you will be presented with a list box with all the information in it. When you exit the list box, you will be presented with a dialog box similar to the following:



Either enter a file name to save the information into a file or press ESC to abort the save. If a file name was entered and Enter was pressed the following processing dialog box will be displayed as the log file is being saved:



A log file will be created for you if one does not already exist. If the file does exist, then the current log will be appended to the end of the previous one.

The five basic testing modes are as follows:

System Configuration

The System Configuration menu leads you to two options.



System Information

The System Information menu will display a dialog box similar to the following:

```
System Information
OS/2 OEM Name       :IBM Corp.
OS/2 Internal Revision :20.30
PC Manufacturer     :
PC Model            :
PC Rom Revision     :
PC Display Code     :8
BIOS PC Model       :252
BIOS PC Sub Model   :1
                   :AT Model 319/339
BIOS Bios Revision  :0
BIOS Bios Flag      :0
Video Type          :10
SVGA Adapter        :2
SVGA Chip Set       :2
                   :Trident Microsystems 8900
SVGA Video Memory   :1048576
Memory (below 1Mb)  :640
Memory (above 1Mb)  :31744
```

It lists various facts about your OS/2 system. The items which are displayed are as follows:

OS/2 OEM Name	This is the name of an OEM. If it is not an IBM version of the OS/2 operating system, it may indicate that extra functionality has been provided.
OS/2 Internal Revision	The OS/2 version number.
PC Manufacturer	Name of manufacturer from ROM (if available)
PC Model	Model number from ROM (if available)
PC Rom Revision	ROM revision number (if available)

PC Display Code	Display Combination Code																														
	Code returned from INT 10h (AH=1Ah).																														
	<table> <thead> <tr> <th>Value</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>No display.</td> </tr> <tr> <td>1</td> <td>Monochrome with 5151 (monochrome) monitor.</td> </tr> <tr> <td>2</td> <td>CGA with 5153.4 (colour) monitor.</td> </tr> <tr> <td>3</td> <td>RESERVED.</td> </tr> <tr> <td>4</td> <td>EGA with 5153/4 (colour) monitor.</td> </tr> <tr> <td>5</td> <td>EGA with 5151 (monochrome) monitor.</td> </tr> <tr> <td>6</td> <td>PGA with 5175 (colour) monitor.</td> </tr> <tr> <td>7</td> <td>VGA with analog monochrome monitor. Except IBM PS/2 Models 25 and 30.</td> </tr> <tr> <td>8</td> <td>VGA with analog colour monitor. Except IBM PS/2 Models 25 and 30.</td> </tr> <tr> <td>10</td> <td>RESERVED.</td> </tr> <tr> <td>11</td> <td>MCGA with analog monochrome monitor. IBM PS/2 Models 25 and 30 only.</td> </tr> <tr> <td>12</td> <td>MCGA with analog colour monitor. IBM PS/2 Models 25 and 30 only.</td> </tr> <tr> <td>13-254</td> <td>RESERVED.</td> </tr> <tr> <td>255</td> <td>Unknown monitor type.</td> </tr> </tbody> </table>	Value	Description	0	No display.	1	Monochrome with 5151 (monochrome) monitor.	2	CGA with 5153.4 (colour) monitor.	3	RESERVED.	4	EGA with 5153/4 (colour) monitor.	5	EGA with 5151 (monochrome) monitor.	6	PGA with 5175 (colour) monitor.	7	VGA with analog monochrome monitor. Except IBM PS/2 Models 25 and 30.	8	VGA with analog colour monitor. Except IBM PS/2 Models 25 and 30.	10	RESERVED.	11	MCGA with analog monochrome monitor. IBM PS/2 Models 25 and 30 only.	12	MCGA with analog colour monitor. IBM PS/2 Models 25 and 30 only.	13-254	RESERVED.	255	Unknown monitor type.
Value	Description																														
0	No display.																														
1	Monochrome with 5151 (monochrome) monitor.																														
2	CGA with 5153.4 (colour) monitor.																														
3	RESERVED.																														
4	EGA with 5153/4 (colour) monitor.																														
5	EGA with 5151 (monochrome) monitor.																														
6	PGA with 5175 (colour) monitor.																														
7	VGA with analog monochrome monitor. Except IBM PS/2 Models 25 and 30.																														
8	VGA with analog colour monitor. Except IBM PS/2 Models 25 and 30.																														
10	RESERVED.																														
11	MCGA with analog monochrome monitor. IBM PS/2 Models 25 and 30 only.																														
12	MCGA with analog colour monitor. IBM PS/2 Models 25 and 30 only.																														
13-254	RESERVED.																														
255	Unknown monitor type.																														
BIOS PC Model	Machine model byte. As returned from the BIOS.																														
BIOS PC Sub Model	Machine sub model byte. As returned from the BIOS.																														
BIOS Bios Revision	BIOS revision level.																														
BIOS Bios Flag	BIOS/ABIOS Flags																														
	<table> <tbody> <tr> <td>BIT 0</td> <td>0x01</td> <td>A value of 1 indicates that ABIOS is present.</td> </tr> <tr> <td>BIT 1-15</td> <td></td> <td>Reserved.</td> </tr> </tbody> </table>	BIT 0	0x01	A value of 1 indicates that ABIOS is present.	BIT 1-15		Reserved.																								
BIT 0	0x01	A value of 1 indicates that ABIOS is present.																													
BIT 1-15		Reserved.																													

Video Type

Video Adaptor Type

Bit 0	0x01	MPA
Bit 1	0x02	CGA
Bit 2	0x04	EGA
Bit 3	0x08	VGA
Bits 4-7	0x?0	Reserved

SVGA Adaptor

Adaptor Type

Returns the code value of the chip set manufacturer, and has one of the following values:

0	Indeterminate Chip Set
1	Headland Technology, Inc.
2	Trident Microsystems
3	Tseng Lab
4	Western Digital
5	ATI Technologies
6	IBM
7	Cirrus Logic
8	S3
9	Not Used
10	Weitek

SVGA Chip Set

Chip Type

Returns the chip type and has a value of 1 through 4 according to the specific manufacturer. Refer to the following table for more information:

Indeterminate Chip Set		0	0
Headland Technology	HT205	1	1
	HT208	1	2
	HT209	1	3
Trident Micro systems	8800	2	1
	8900	2	2
Tseng Labs	ET3000	3	1
	ET4000	3	2
	ET4000W32	3	3
	ET4000W32I	3	4
	ET4000W32IB	3	5
	ET4000W32IC	3	6
	ET4000W32PA	3	7
	ET4000W32PB	3	8
	ET4000W32PC	3	9
	ET4000W32ID	3	10
	ET4000W32PD	3	11
	ET4000W32PX	3	12
	Western Digital	PVGA1A	4
WD90C00		4	2
WD90C11		4	3
WD90C30		4	4
WD90C26		4	5
WD90C27		4	6
WD90C31		4	7
WD90C24		4	8
WD90C33		4	9
ATI Technologies		18800	5
	28800	5	2
	38800	5	3
	68800	5	4
	88800	5	5
	88800CT	5	6
IBM	VGA-256C	6	0
	IBMSVGA	6	1

Cirrus Logic	GD54207	1	
	GD54227	2	
	GD54237	3	
	GD54267	4	
	GD54287	5	
	GD54297	6	
	GD543x	7	7
	GD54347	8	
S3	86C805	8	1
	86C928	8	2
	86C911	8	3
	86C864	8	4
	86C964	8	5
Weitek	P9000	10	1
	W5186	10	2
	W5286	10	3
	P9100	10	4

SVGA Video Memory Returns the detected memory size of the video adaptor.

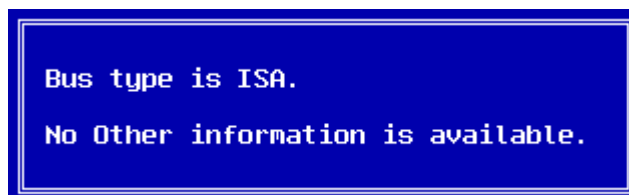
Memory (Below 1Mb) The amount of random access memory below the 1 megabyte address. The value returned is in kilobytes.

Memory (Above 1Mb) The amount of random access memory above the 1 megabyte address. The value returned is in kilobytes.

Bus Information

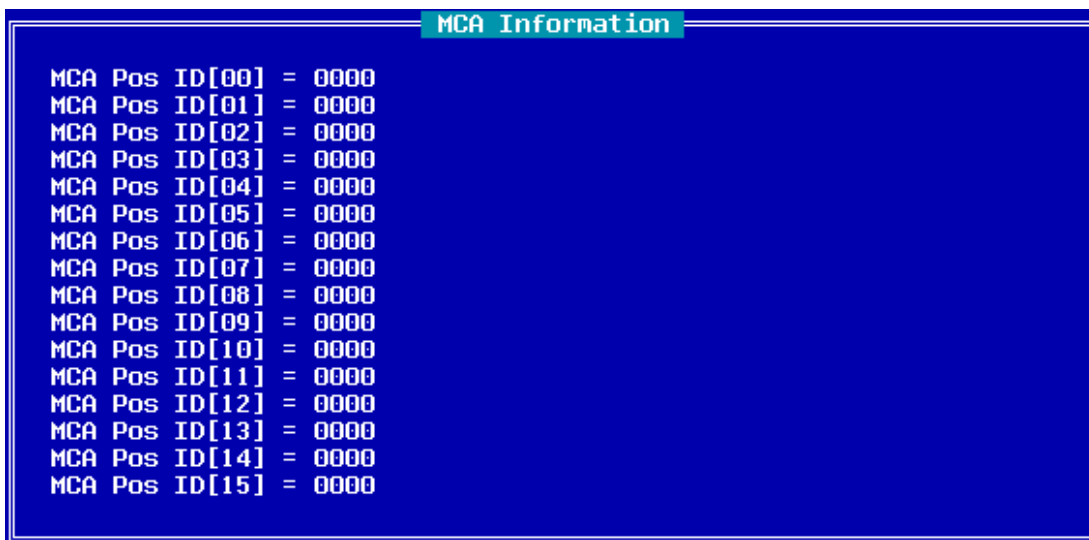
The Bus Information menu will display one of the following dialog boxes. The three bus types which are recognised are: ISA, MCA and EISA.

For ISA based machines you will see this dialog box:



PCI and VLB machines will also appear as ISA.

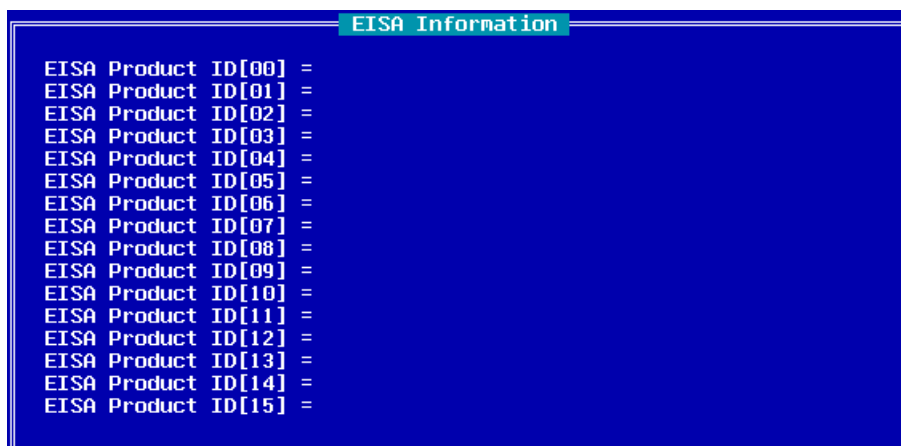
For MCA based machines you will see a dialog box similar to the following:



```
MCA Information
MCA Pos ID[00] = 0000
MCA Pos ID[01] = 0000
MCA Pos ID[02] = 0000
MCA Pos ID[03] = 0000
MCA Pos ID[04] = 0000
MCA Pos ID[05] = 0000
MCA Pos ID[06] = 0000
MCA Pos ID[07] = 0000
MCA Pos ID[08] = 0000
MCA Pos ID[09] = 0000
MCA Pos ID[10] = 0000
MCA Pos ID[11] = 0000
MCA Pos ID[12] = 0000
MCA Pos ID[13] = 0000
MCA Pos ID[14] = 0000
MCA Pos ID[15] = 0000
```

The Pos ID's of each adaptor will be displayed.

For EISA based machines you will see a dialog box similar to the following:

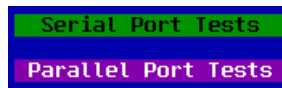


```
EISA Information
EISA Product ID[00] =
EISA Product ID[01] =
EISA Product ID[02] =
EISA Product ID[03] =
EISA Product ID[04] =
EISA Product ID[05] =
EISA Product ID[06] =
EISA Product ID[07] =
EISA Product ID[08] =
EISA Product ID[09] =
EISA Product ID[10] =
EISA Product ID[11] =
EISA Product ID[12] =
EISA Product ID[13] =
EISA Product ID[14] =
EISA Product ID[15] =
```

The Product ID's of each EISA adaptor will be displayed.

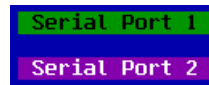
Communication Port Tests

The Communications Port Tests menu gives you the choice of testing either the serial ports or the parallel ports in your system.



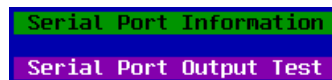
Serial Port Tests

Choosing the Serial Port Tests option allows you to test the serial ports. A serial port must be chosen in order to be tested.



The actual number of serial ports displayed will be dependant on the number of serial ports in your system.

Once a particular serial port has been chosen for testing, the choice of either reporting on the current serial port status or testing the serial port output lines is presented to you:



Serial Port Information

The Serial Port Information menu will present a dialog box similar to the following:

```
Serial Port Information (COM1)
Current Information:
Bit Rate: 1200 Data Bits: 7 Stop Bits: 0 (1 Stop Bit) Parity: 2 (Even)
Comm Status : 00
Line Status : 00
Modem Control Out: 03 (DTR ON) (RTS ON)
Modem Control In : 00 (CTS OFF) (DSR OFF) (RI OFF) (DCD OFF)
Extended Information:
Current Bit Rate: 1200 Fractional Bit Rate: 0
Minimum Bit Rate: 2 Fractional Bit Rate: 0
Maximum Bit Rate: 57600 Fractional Bit Rate: 0
Buffer Information:
Bytes in Transmit Queue: 0 Transmit Queue Size: 2048
Bytes in Receive Queue: 0 Receive Queue Size: 5120
Device Control Block Information:
Write Timeout: 60.01 seconds Read Timeout: 60.01 seconds
Control Hand Shake: 01 Flow Replace: 40 Timeout: 02
Error Replacement Char: 0 Break Replacement Char: 0
XON Char: 17 XOFF Char: 19
Enhanced Mode Information:
Enhanced Flag: 0
```

The Serial Port Information dialog box lists the current status of your com port. It is divided into a number of groups.

Current Information:

Bit Rate The Bit Rate field is a binary integer representing the actual bit rate (bits-per-second) which the physical driver uses to set the bit rate of the COM device. The recommended bit rate values are:

110, 150, 300, 600, 1200, 1800, 2000, 2400, 3600, 4800, 7200, 9600, 19200, 38400, 57600 and 115200.

Data Bits 00h-04h Reserved
05h 5 data bits
06h 6 data bits
07h 7 data bits (initial value)
08h 8 data bits
09h-FFh Reserved

Stop Bits	00h	1 stop bit (initial value)
	01h	1.5 stop bits (valid with 5-bit WORD length only)
	02h	2 stop bits (not valid with 5-bit WORD length)
	03h-FFh	Reserved
Parity	00h	No parity
	01h	Odd parity
	02h	Even parity (initial value)
	03h	Mark parity (parity bit always 1)
	04h	Space parity (parity bit always 0)
	05h-FFh	Reserved
Comm Status	Please consult the OS/2 programming documentation for the full description of this value.	
Line Status	Returned as bit-significant values. If the bit is 1, the condition is TRUE. If the bit is 0, the condition is FALSE. The number at the beginning of the description is the bit position number. The bit positions go from least to most significant.	
	Bit 0	WRITE request packets in progress or queued
	Bit 1	Data in the physical device driver transmit queue
	Bit 2	Transmit hardware is currently transmitting data
	Bit 3	Character waiting to be transmitted immediately
	Bit 4	Waiting to automatically transmit an XON
	Bit 5	Waiting to automatically transmit an XOFF
	Bit 6	Undefined
	Bit 7	Undefined
Modem Control Out	If a bit has a value of 1, the condition is on. If a bit has a value of 0, the condition is off.	
	Bit 0	Data Terminal Ready (DTR)
	Bit 1	Request To Send (RTS)
	Bits 2-7	Undefined

Modem Control In If a bit has a value of 1, the condition is on. If a bit has a value of 0, the condition is off.

Bit 0-3 Undefined

Bit 4 Clear To Send (CTS)

Bit 5 Data Set Ready (DSR)

Bit 6 Ring Indicator (RI)

Bit 7 Data Carrier Detect (DCD)

Extended Information:

Current Bit Rate The binary integer representing the actual bit rate in bits-per-second set for a COM port.

Fractional Bit Rate The binary integer representing the fraction of the actual current bit rate set for a COM port.

Minimum Bit Rate The binary integer representing the minimum bit rate (in bits-per-second) port for a COM port.

Fractional Bit Rate The binary integer representing the fraction of the minimum bit rate supported for a COM port.

Maximum Bit Rate The binary integer representing the maximum bit rate (in bits-per-second) supported for a COM port. Depending on overall system overhead and the electrical characteristics of the hardware cables and serial device adaptor type, the actual value of the maximum bit rate supported might be lower than this.

Fractional Bit Rate The binary integer representing the fraction of the maximum bit rate supported for a COM port.

Buffer Information:

Bytes in Transmit Queue	Binary integer with the number of characters ready to be transmitted in the physical device driver transmit queue. This is a memory buffer between the memory pointed to by the WRITE request packet and the transmit hardware for this COM device.
Transmit Queue Size	Binary integer with the size of the physical device driver transmit queue.
Bytes in Receive Queue	Binary integer with the number of received characters in the device driver receive queue. This is a memory buffer between the memory pointed to by the READ request packet and the receive hardware for this COM device.
Receive Queue Size	Binary integer with the size of the physical device driver receive queue.

Device Control Block Information:

Write Timeout	Specifies the time period used for Write Timeout processing.
Read Timeout	Specifies the time period used for Read Timeout processing.

Control Handshake Has the following bits:

Bits 0-1 DTR Control mode. Has the following:

	Bit 1	Bit 0	Description
	0	0	Disable
	0	1	Enable
	1	0	Input handshaking
	1	1	Invalid input. Results in a general failure error.
Bit 2			Reserved (returned as 0)
Bit 3			Enable output handshaking using CTS
Bit 4			Enable output handshaking using DSR
Bit 5			Enable output handshaking using DCD
Bit 6			Enable input sensitivity using DSR
Bit 7			Reserved (returned as 0)

Flow Replace Has the following bits:

Bit 0			Enable Automatic Transmit Flow Control (XON/XOFF)
Bit 1			Enable Automatic Receive Flow Control (XON/XOFF)
Bit 2			Enable error replacement character
Bit 3			Enable null stripping (remove null bytes)
Bit 4			Enable break replacement character
Bit 5			Automatic Receive Flow Control
			0 = Normal
			1 = Full-Duplex

Bits 6-7 RTS Control mode. Has the following:

	Bit 7	Bit 6	Description
	0	0	Disable
	0	1	Enable
	1	0	Input handshaking
	1	1	Toggling on transmit

Timeout

Has the following bits:

Bit 0 Enable Write Infinite Timeout processing

Bits 1-2 Read Timeout processing. Has the following:

Bit 2	Bit 1	Description
0	1	Normal Read Timeout processing
1	0	Wait-For-Something, Read Timeout processing
1	1	No-Wait, Read Timeout processing

Bits 3-4 Extended Hardware Buffering. Has the following:

Bit 4	Bit 3	Description
0	0	Not supported
0	1	Extended Hardware Buffering Disabled
1	0	Extended Hardware Buffering Enabled
1	1	Automatic Protocol Override

Bits 5-6 Receive Trigger Level. Has the following:

Bit 6	Bit 5	Description
0	0	1 character
0	1	4 characters
1	0	8 characters
1	1	14 characters

Bit 7 Transmit Buffer Load Count

0 = 1 character
1 = 16 characters

Error Replacement Char

This is the error replacement character. Please consult the OS/2 programming documentation for a complete description of this value.

Break Replacement Char

This is the break replacement character. Please consult the OS/2 programming documentation for a complete description of this value.

XON Char

This is the XON character. Please consult the OS/2 programming documentation for a complete description of this value.

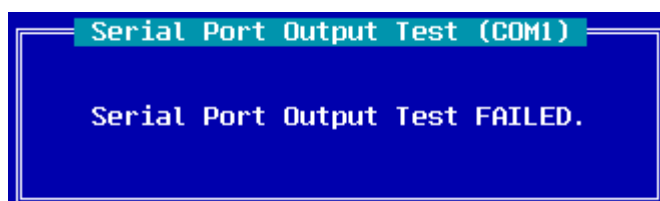
XOFF Char This is the XOFF character. Please consult the OS/2 programming documentation for a complete description of this value.

Enhanced Mode Information:

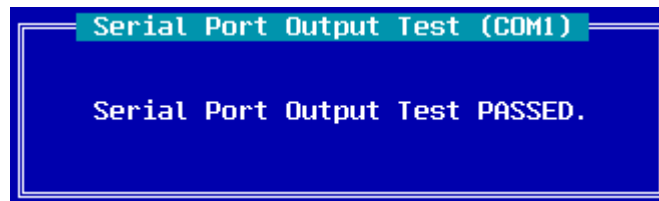
Enhanced Flag Bit 0 Enhanced mode supported by hardware.
 Bit 1 Enable the Enhanced mode (default)
 Bits 2-3 DMA Receive Operation request. Has the following:
 Bit 3 Bit 2 Description
 0 0 Disable DMA Receive Capability
 0 1 Enable DMA Receive Capability (Default)
 1 0 Dedicate a DMA channel to Receive operation
 1 1 Reserved
 Bits 4-5 DMA Transmit operation request. Has the following:
 Bit 5 Bit 4 Description
 0 0 Disable DMA Transmit Capability
 0 1 Enable DMA Transmit Capability (Default)
 1 0 Dedicate a DMA channel to Transmit operation
 1 1 Reserved
 Bit 6 Receive operation in DMA mode
 Bit 7 Transmit operation in DMA mode

Serial Port Output Test

If the serial port output test failed you will be presented with this dialog box:



If the serial port test passes, you will see a similar dialog box to this:



As there is no loop back connector supplied, only a minimal test can be performed.

Please note: As the system boots OS/2 performs significant testing of the system. If there are problems with the comm ports, the COM.SYS driver will not install and you will be notified of an error.

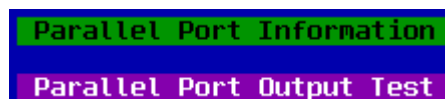
Parallel Port Tests

Choosing the Parallel Port Tests option allows you to test the serial ports. A parallel port must be chosen in order to be tested.



The actual number of parallel ports displayed will be dependant on the number of parallel ports in your system.

Once a particular parallel port has been chosen for testing, you are presented with the choice of either reporting on the current parallel port status or testing the parallel port output lines is presented to you:



Parallel Port Information

Choosing the Parallel Port Information menu will present a dialog box similar to the following:

```

Parallel Port Information (LPT1)
Frame Control Chars Per Line :80
Frame Control Lines Per Inch :6
Infinite Retry Flag          :1
Printer Status                :90
                             : Selected (On Line).
Current Font Code Page       :437
Current Font Font ID         :0
  
```

Frame Control Chars Per Line Valid characters per line (CPL) value. For the SBCS code page, valid values are 80 and 132.

Frame Control Lines Per Inch Valid lines per inch (LPI) value. For the SBCS code page, valid values are 6 and 8.

Infinite Retry Flag

0	Disable infinite retry
1	Enable infinite retry

Printer Status

Bit 7	0x80	1 = Not Busy
Bit 6	0x40	1 = Acknowledge
Bit 5	0x20	1 = Out of Paper
Bit 4	0x10	1 = Selected
Bit 3	0x08	1 = I/O Error
Bit 2	0x04	Unused
Bit 1	0x02	Unused
Bit 0	0x01	1 = Timeout

Current Font Code Page This is the currently active code page.

0000H If the Code Page value and Font ID are returned as zero, the printer is set to the hardware default code page and font.

0001H-FFFFH Valid code page numbers.

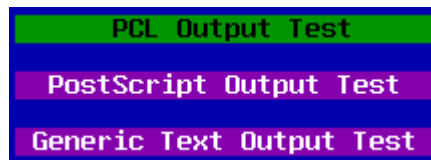
Current Font Font ID	This is the ID value of the font which is currently active.	
	0000H	If the Code Page value and Font ID are specified as zero, the printer is set to the hardware default code page and font. If Font ID is zero and code page is non zero, no error is returned if any Font ID is available for the specified code page.
	0001H-FFFFH	Valid Font ID numbers; font types defined by the font file definitions for down loadable fonts. For cartridge fonts, Font IDs are the numbers on the cartridge label, and are also entered in the DEVINFO statement for the printer.

Note:

If the chosen parallel port has been assigned to a network printer, the information displayed will be that of the remote printer.

Parallel Port Output Test

Choosing the Parallel Port Output Test presents you with this menu.



Choosing either the PCL Output Test (for HP LaserJet and compatible printers) or the PostScript Output Test will print the WarpSpeed Computers logo. The Generic Text Output Test will print the following test lines:

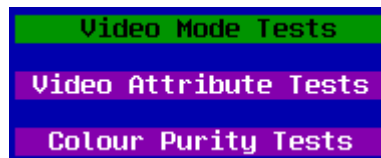
WarpSpeed Computers Diagnostic Test Print

Generic Text

Successfully printing the files indicates that both the printer and spooler sub systems are functioning correctly.

Video Tests

Choosing the Video Mode Tests displays the following menu:



Video Mode Tests

Choosing the Video Mode Tests menu will present the following selection menu:

Mode	00h	Text	40x25x2
Mode	01h	Text	40x25x16
Mode	02h	Text	80x25x2
Mode	03h	Text	80x25x16
Mode	04h	Graphics	320x200x4
Mode	05h	Graphics	320x200x4
Mode	06h	Graphics	640x200x2
Mode	07h	Text	80x25x2
Mode	0Dh	Graphics	320x200x16
Mode	0Eh	Graphics	640x200x16
Mode	0Fh	Graphics	640x350x2
Mode	10h	Graphics	640x350x16
Mode	11h	Graphics	640x480x2
Mode	12h	Graphics	640x480x16
Mode	13h	Graphics	320x200x256

Choose the mode which you wish to test.

For text modes, the complete character set will be displayed. The character set which is displayed is dependent on the current code page which has been prepared for your system. The current code page can be displayed by entering the CHCP command from a command line.

The standard US code page, 437 looks like this:



The multilingual code page, 850 looks like this:

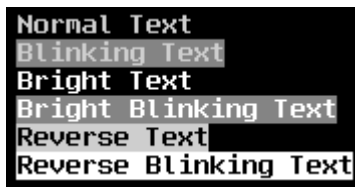


For graphics modes, a grid is displayed. The grid should be straight and even. There should be no breaks in any of the lines.

Video Attribute Tests

Choosing the Video Attribute Tests menu will present you with the same menu as for the Video Mode Tests. Choose the mode which you wish to test.

For text modes you will be displayed with the six available text attribute types.



Each line should appear different and as described above.

For graphics modes, colour bars are drawn. A bar of each colour is drawn. You should see colour bars of equal width and of equal colour intensity.

Colour Purity Tests

The colour purity tests are designed to test the display attached to your system. Three solid screens of red, green and blue are displayed consecutively. The colour should appear to be solid and consistent through out the display area.

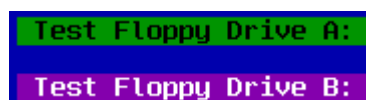
Hard Disk Tests

Selecting the Hard Disk Tests menu will display all local hard disks available for testing.

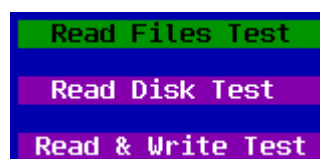


Floppy Disk Tests

Selecting the Floppy Disk Tests menu will display all floppy drives attached to your system.

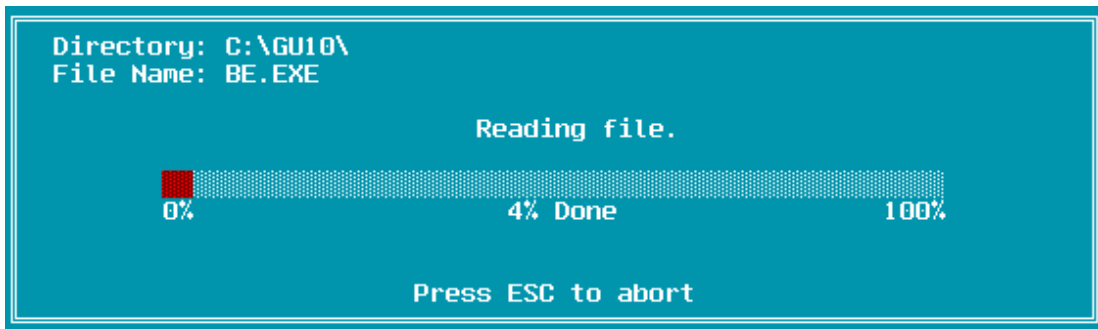


Selecting either a hard disk or floppy disk to test will present the following menu:



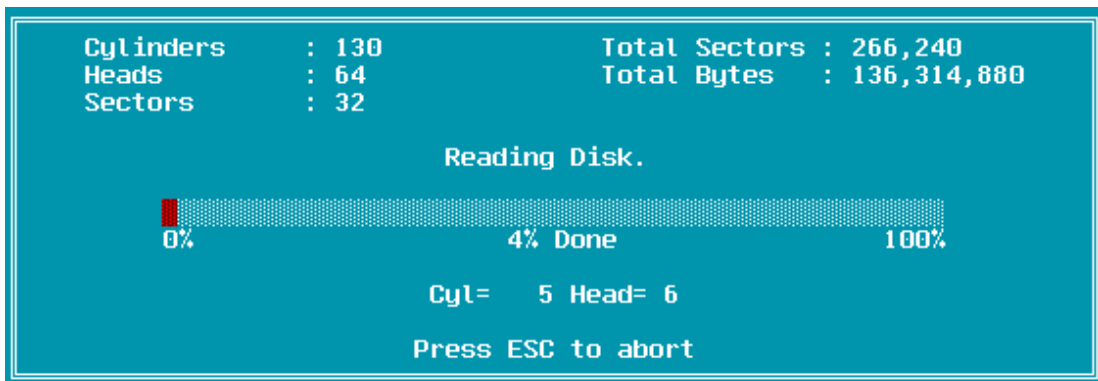
Read Files Test

The Read Files Test test reads all files on the selected drive. As each file is read it is added to the log. Any errors will be displayed and added to the log. As the file read test progresses the following bar graph will be displayed. The overall transfer rate of all files that have been read will also be added to the log.



Read Disk Test

The Read Disk Test reads the selected disk at a sector level. One complete track is read at a time, head by head, starting at cylinder zero up to the maximum number of cylinders on the disk. As the test progresses, a bar graph indicating progress is displayed.

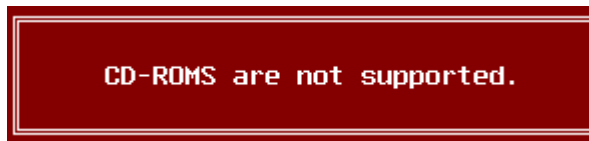


As the test proceeds any errors will be displayed and added to the log.

Read & Write Test

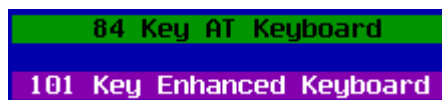
The Read & Write Test performs a non-destructive test on the selected disk. The disk is first read (to save the information there). Then the entire track is written with 0's. The writing of the 0's is verified. Then the entire track is written with FF's. The writing of the FF's is verified. Finally the original data is replaced. As the test proceeds, a dialog box similar to the Read Disk Test is displayed.

If a local CD-ROM has been selected for either the Read Disk Test or the Read & Write Test, the following dialog box will be displayed and no tests will be performed on the drive.

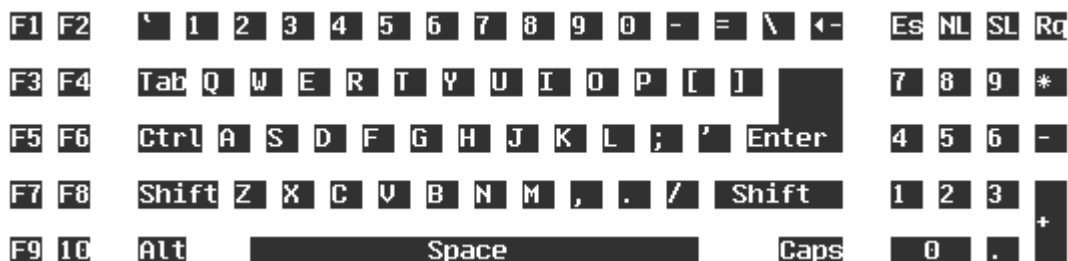


Keyboard Tests

The Keyboard Tests menu gives you the option of testing either an 84 key AT style keyboard or a 101 key enhanced style keyboard. You will be presented with this menu:



As you select which keyboard you wish to test, the keyboard layout is displayed at the top of the screen. The 84 key AT keyboard layout looks like this:



The 101 enhanced keyboard layout looks like this:

```
Esc  F1 F2 F3 F4  F5 F6 F7 F8  F9 10 11 12  Prt Scr Pau

` 1 2 3 4 5 6 7 8 9 0 - = <-BS  Ins Hom PgU  NL / * -
Tab Q W E R T Y U I O P [ ] \  Del End PgD  7 8 9
Caps A S D F G H J K L ; ' Enter  4 5 6
Shift Z X C V B N M , . / Shift  ↑  1 2 3
Ctrl Alt  Space  Alt Ctrl  ← ↓ →  0 .
```

Select the keyboard which you wish to test by choosing the appropriate menu item. To test the keyboard all you have to do is to press every key. No particular order is required. To abort the test, simply press any key five times. If all keys are pressed, the test has passed, if keys were missed, every key that was missed is added to the log and the test is failed.

Example

DIAG

DIAG has no switches.

DiskEdit - Disk Editor

Summary

DiskEdit allows you to edit disks.

Icon



Command Line Format

Usage: DiskEdit {Switches} <Drive Spec>

Switches

- | | |
|--------------|--|
| -nolock | No disk lock is required.
Use this option to view disks that are locked.
This option turns disk writes off. |
| -allowwrites | Allow disk writes.
The -nolock options turns disk writes off,
this option turns them back on. This option
MUST be specified after the -nolock switch.
WARNING!
Use of this option is NOT recommended. |
| -c | Specify an initial Cluster. |
| -i | Toggle the Initial edit mode. |
| -l | Specify an initial Logical Sector. |
| -p<n> | Physical disk editing mode.
(The default mode is Logical disk editing mode.) |
| -r | Force RAW disk editing mode. |
| -u | Automatic update of HPFS386 check sum values.
(Super and Spare Blocks only). |
| -on | Run on <Machine Name>. |

Description

DiskEdit is a program which allows you to edit both floppy and hard disks. It enables you to edit hard disks in either logical or physical disk modes. The physical disk mode allows you to edit a hard disk on an absolute sector basis. You will need to use this mode if you wish to edit your partition table. The logical disk mode only allows you to edit sectors within a logical disk.

The default editing mode allows you to edit sectors in hexadecimal or ASCII. This is known as RAW mode. DiskEdit will display a complete sectors' worth of data at a time for you to edit. It will change the screen mode to 35 rows to accommodate the amount of data which needs to be displayed. If the screen mode can not be changed, DiskEdit will default to 25 rows. However it can not display a complete sectors' worth of data, and you will have to scroll the screen to see all of the data.

If you enter DiskEdit with no parameters it will default to editing the current disk in logical mode. If you wish to edit the total disk, which may span multiple logical partitions, use the `-p` switch to force DiskEdit to edit in physical disk mode.

DiskEdit uses a DLL to do most of the work when editing a disk. DiskEdit itself manages the higher level functions as well as which particular DLL is called. DiskEdit searches for a DLL with a name of the form `DE<FSName>.DLL` - where `FSName` is the name of the file system (FAT, HPFS and so on). If DiskEdit can not find the appropriate DLL (`DEFAT.DLL`, `DEHPFS.DLL` and so on), `DERAW.DLL` will be loaded. You may also force DiskEdit to load `DERAW.DLL` by using the `-r` switch. The DLL contains the lower level functions, such as the reading and writing of the disk as well as the editing functions. The first thing which DiskEdit does is to load the appropriate DLL and display its version information. It will be one of the four `DE*.DLL`'s supplied.

```
The Graham Utilities for OS/2 V2.00.  
Disk Editor Module  
DERAW.DLL  
Version 2.00  
31/08/96
```

This is the initialisation dialog box for DERAU.DLL. It is used for logical disk editing.

```
The Graham Utilities for OS/2 V2.00.  
Disk Editor Module  
DEPHYS.DLL  
Version 2.00  
31/08/96
```

This is the initialisation dialog box for DEPHYS.DLL. It is used for physical disk editing.

```
The Graham Utilities for OS/2 V2.00.  
Disk Editor Module  
DEPIPE.DLL  
Version 2.00  
31/08/96
```

This is the initialisation dialog box for DEPIPE.DLL. It is used for remote disk editing.

```
The Graham Utilities for OS/2 V2.00.  
Disk Editor Module  
DEHPFS.DLL  
Version 2.00  
31/08/96
```

This is the initialisation dialog box for DEHPFS.DLL. It is used for editing HPFS partitions in their native formats.

```
The Graham Utilities for OS/2 V2.00.  
Disk Editor Module  
DEFAT.DLL  
Version 2.00  
31/08/96
```

This is the initialisation dialog box for DEFAT.DLL. It is used for editing FAT disks in their native formats.

Please consult Appendix I - DiskEdit Modules, for further information on all of the DE*.DLL's and the editing modes which they provide.

After doing this, DiskEdit attempts to lock the disk for exclusive access. This is so that no other processes can access the disk while you are editing it. It has been done to protect your data. DiskEdit has the ability to change data which may be in use by other applications. These other applications will not be notified that the data has changed, and thus data corruption is likely. If DiskEdit can not obtain exclusive access to a disk, you will be prompted with an error dialog box similar to the following:

```
SYS0108: The disk is in use or locked by  
another process.
```

for logical disk modes, and:

```
SYS0033: The process cannot access the file because
another process has locked a portion of the file.
```

for physical disk modes.

If you only have one disk (either physical or logical) you will see one of these error messages. This is because OS/2 itself is using the disk. If you wish to edit the disk safely, you should reboot OS/2 off a floppy and then use DiskEdit.

If you wish to simply view the data on a disk and not edit it, you may use the "-nolock" switch. This switch allows you full read access to the disk, but it disables all editing and writing functions. If you attempt any of these functions you will hear a low tone warning you of an illegal function.

If you have the absolute need to edit an unlocked disk, you may enter the "-nolock" switch followed by the "-allowwrites" switch. They must be specified in that order. As a reminder you will be given an audible alarm and be presented with the following dialog box:

```
!!!! WARNING !!!!

The NO DISK LOCK OPTION
has been selected. This
means that you can modify
data that could be in use
by other processes.

Data corruption is likely.

The use of this option is
NOT RECOMMENDED
```

Press any key to continue.

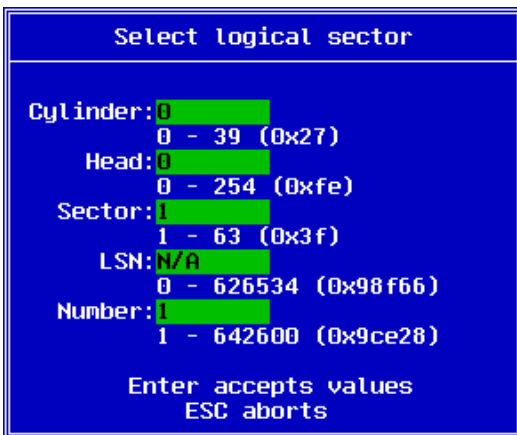
The next thing that DiskEdit does is to validate the disk parameter values. If there are any discrepancies between the actual and reported values, DiskEdit will warn you with the following dialog box:

```
Warning! The ACTUAL values and RECOMMENDED values do not
match. The ACTUAL values will be used.
```

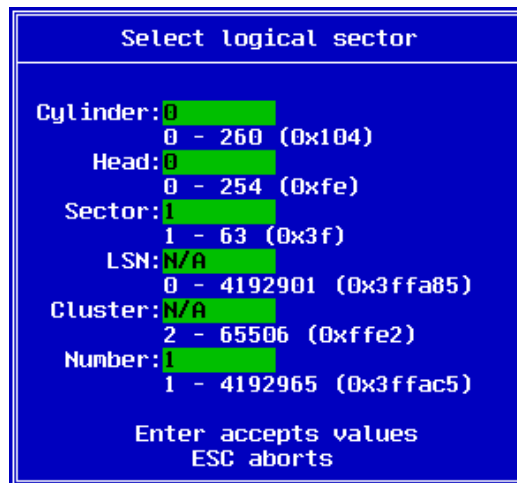
Recommended Values	Drive A:	Actual Values
512	bytes per sector	512
1	sectors per cluster	1
1	reserved sectors	1
2	number of FATs	4
224	root directory entries	224
2880	number of sectors	2880
F0	media descriptor	F0
9	sectors per FAT	9
18	sectors per track	18
2	number of heads	2
0	hidden sectors	0
0	large sectors	0
80	number of cylinders	80
2	device attributes	2
7	device type	7

You should not continue beyond this point. You should take actions to correct the problem. It probably means that you should back your data up and reformat the disk. In the above example, the number of FATs is incorrect. The disk reports that it has 4 FATs, but OS/2 recommends that it should only have two.

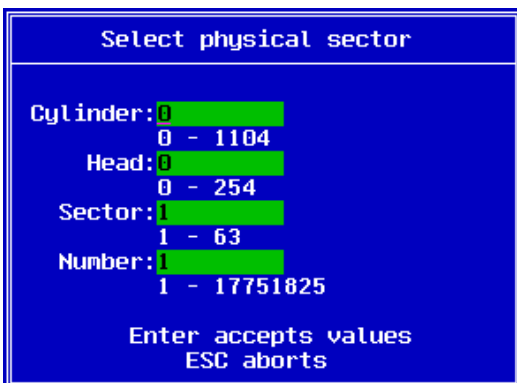
If disk has passed all of these checks, DiskEdit will now prompt you with the sector selection dialog box similar to one of these:



This is the sector selection dialog box for the DERAW, DEHPFS and DEPIPE DLL's.



This is the sector selection dialog box for the DEFAT DLL.



This is the sector selection dialog box for the DEPHYS DLL.

It is here that you enter the Cylinder, Head and Sector at which you wish to start editing. The Number field is used to specify the number of sectors from the point at which you wish to conduct search, file read and file write operations.

Use the Up, Down and Tab keys to move between fields. The Enter key accepts the entered values. The Escape key will exit the entry of values and DiskEdit will exit.

If you pressed the Enter key, DiskEdit will read the sector and display it to you. It may look similar to this:

```

Logical Disk C: Cylinder 0000 Head 00 Sector 01 RAW No Mask Byte Offset 000
00000000 fa b8 30 00 8e d0 bc 00 01 fb fc b8 00 00 8e d8
00000010 8e c0 be 00 7c bf 00 7e b9 00 02 f3 a5 b8 22 7e
00000020 50 c3 be 92 7e bb be 7d 33 c9 80 3f 80 75 08 0b
00000030 c9 75 4d 8b cb eb 05 80 3f 00 75 44 83 c3 10 81
00000040 fb fe 7d 7c e5 0b c9 75 02 cd 18 8b d9 8b 4f 02
00000050 8a 77 01 b2 80 bb 00 7c bf 05 00 b8 00 00 cd 13
00000060 b8 01 02 cd 13 73 08 4f 7f f1 be cd 7e eb 11 be
00000070 0f 7f 26 81 3e fe 7d 55 aa 75 05 ea 00 7c 00 00
00000080 33 db eb 02 cd 10 b4 0e ac 0a c0 75 f7 fb eb fe
00000090 38 00 54 68 65 20 70 61 72 74 69 74 69 6f 6e 20
000000A0 74 61 62 6c 65 20 6f 6e 20 74 68 65 20 73 74 61
000000B0 72 74 75 70 20 64 72 69 76 65 20 69 73 20 69 6e
000000C0 63 6f 72 72 65 63 74 2e 0d 0a 00 3f 00 54 68 65
000000D0 20 6f 70 65 72 61 74 69 6e 67 20 73 79 73 74 65
000000E0 6d 20 63 61 6e 6e 6f 74 20 62 65 20 6c 6f 61 64
000000F0 65 64 20 66 72 6f 6d 20 74 68 65 20 73 74 61 72
00000100 74 75 70 20 64 72 69 76 65 2e 0d 0a 00 39 00 54
00000110 68 65 20 6f 70 65 72 61 74 69 6e 67 20 73 79 73
00000120 74 65 6d 20 69 73 20 6d 69 73 73 69 6e 67 20 66
00000130 72 6f 6d 20 74 68 65 20 73 74 61 72 74 75 70 20
00000140 64 72 69 76 65 2e 0d 0a 00 00 00 00 00 00 00
00000150 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00000160 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00000170 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00000180 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00000190 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000001A0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000001B0 00 00 00 00 00 00 00 00 cb e1 18 22 00 00 80 01
000001C0 01 00 06 3f 20 45 20 00 00 00 e0 2f 02 00 00 00
000001D0 01 46 05 3f 60 ff 00 30 02 00 00 a8 26 00 00 00
000001E0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000001F0 00 00 00 00 00 00 00 00 00 00 00 00 00 55 aa
Press F1 for Help

```

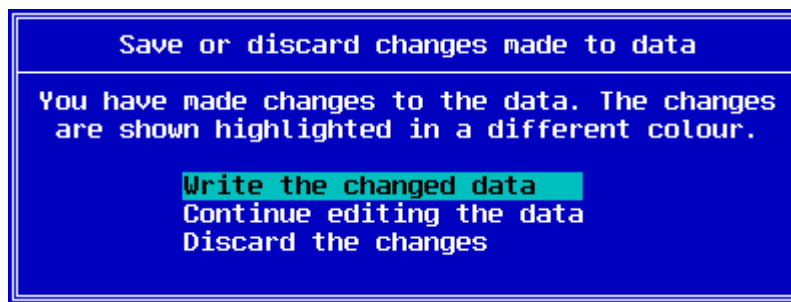
The top status line displays the current editing options and data. It lists whether you are in Logical (Raw, FAT, HPFS or Pipe) or Physical disk mode. The current cylinder, head and sector are displayed in hexadecimal. The current editing display mode is listed along with the current mask state. The byte offset within the current sector may also be displayed.

At this point you are in basic editing mode.

The TAB key switches between hex and ASCII editing modes. When you are editing in hexadecimal, you can only enter hex values ('0'-'9','A'-'F'). In ASCII editing mode you may use all ASCII keys. The cursor control keys are active in both modes. The Page-Up and Page-Down keys move you one sector up or down respectively.

The bytes that have been changed are displayed in yellow. You are always prompted before any data is written back to the disk. You have the option of discarding the data, saving the data or continuing to edit the data.

When you have finished editing the data, and you wish to write the data to disk, simply press the Enter key. You will be presented with the following dialog box:



You may use the Tab, Up or Down Arrow keys to move between the selections, or simply press the highlighted letter to select that option. The Enter key selects the currently selected item.

You will also be prompted with the same dialog box if you attempt to move onto another sector and you have not saved any changes.

-nolock No disk lock is required

Use this option to view disks which are locked. This option also turns disk writes off, so that you can not accidentally modify data on a "live" disk. If you really need to modify data on a disk which you can not lock, use the -allowwrites option.

-allowwrites Allow disk writes

As the -nolock options turns disk writes off, this options turns them back on. This option **MUST** be specified after the -nolock switch.

WARNING!

Use of this option is **NOT** recommended. You could modify data which belongs to a file that is otherwise locked. Modifying data which is in use by another process without that process's knowledge may have disastrous effects.

-c Specify an initial Cluster

This switch enables you to specify an initial starting cluster. With the switch, the initial sector selection dialog box is bypassed. This is only available when editing FAT disks using the DEFAT.DLL module.

-i Toggle the initial editing mode

The default editing mode is RAW hexadecimal. This switch allows you to specify the context sensitive editing mode (if available). This is the equivalent to pressing F6 when the disk editor is first started.

-l Specify an initial Logical Sector

This switch enables you to specify an initial Logical Sector Number. With this switch, the initial sector selection dialog box is bypassed. This is only meaningful with the DERAW, DEHPFS and DEPIPE modules.

-p Physical disk editing mode

The default mode is Logical disk editing mode. This switch changes the operation of DiskEdit so that it accesses complete physical disks, not just logical disks.

Note:

If you only have one primary partition on your disk, the use of this switch is unnecessary.

-r Force RAW disk editing mode

If DiskEdit is to edit a HPFS partition, the DEHPFS.DLL will be loaded to edit it. There may be circumstances where you wish to use the DERAW.DLL instead. This switch forces DERAW.DLL to be loaded.

-u Update HPFS386 check sums

HPFS386 is a version of HPFS that comes with IBM LAN Server. It is optimised for larger disks as used with file servers in LAN environments. With LAN Server Version 4, IBM has introduced checksums of the Super and Spare block for additional data integrity. When the HPFS386 partitions are mounted,

the saved checksum values are compared to the calculated ones. If they do not match then a CHKDSK /F is done - even if the volumes were shut down cleanly. This switch tells DiskEdit to recalculate the appropriate checksums as necessary and save them after your changes have been made.

-on Run on <Machine Name>

This switch forces DEPIPE to be loaded and allows DiskEdit to access disk drives located on remote machines. The <Machine Name> must be specified with a leading "\\\". The machine name of the remote machine will have been specified when the network software on the remote machine was installed.

Note:

DESVR must be running on the remote machine for this option to work.

Consult Appendix H - DESVR for further details.

When you are editing, some of the function keys have the following functions assigned to them.

F1 - Help

The F1 key brings up the help dialog box. It looks like this:

```

                                Disk Editor
                                Edit disks in a hexadecimal format.

While editing a disk, use the following keys:
Page Up           Move to previous sector.
Page Down        Move to next sector.
Enter            Prompt for action.
Tab              Switch between Hex and ASCII edit modes.
Letters          0-9, A-F in Hex mode. All in ASCII mode.
↑,↓,+ or →      Move in the appropriate direction.
F2               Toggle Mask state.
F3               Goto disk location.
F4               Search for data.
F5               Search for next occurrence of data.
F6               Toggle editing mode.
F7               Copy disk to a file.
F8               Copy a file to disk.
F9               Toggle LSN and Cluster Mode
Esc, F10        Quit DiskEdit.
F11             Manually select an editing mode
F12             Jump to a sector.
<Press any key to continue>
```

Press any key and you will be returned to edit mode.

F2 - Toggle mask state

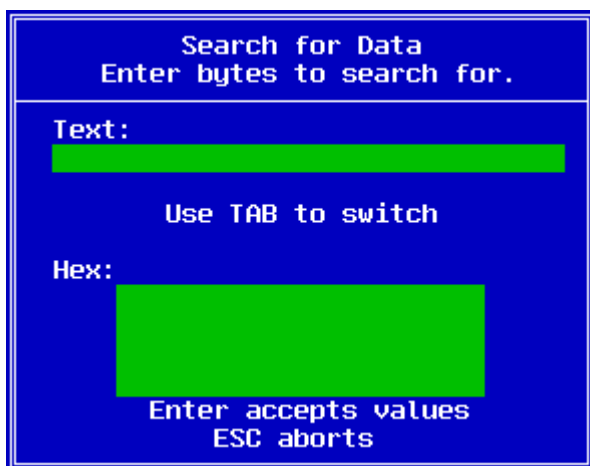
The F2 key toggles the mask state. When masking is on, only normal ASCII characters are displayed. The default is masking off. With masking off, all characters with the exception of the character zero are displayed.

F3 - Go to a location

Use the F3 key to go to a specific location on the disk. The sector selection dialog box will be displayed.

F4 - Search for data

The F4 key is used to search for data. You may enter data in ASCII text form, or in hexadecimal. The Tab key switches between entering data in ASCII and hexadecimal.



A screenshot of a blue dialog box titled "Search for Data" with the instruction "Enter bytes to search for." It contains two input fields: "Text:" with a single-line green bar, and "Hex:" with a multi-line green bar. Between the fields is the text "Use TAB to switch". At the bottom, it says "Enter accepts values" and "ESC aborts".

Pressing Enter will start the search. The Escape key aborts the entry and returns you to the basic editing mode of DiskEdit. DiskEdit will search from the chosen starting point for the number of sectors specified. The search performed is case insensitive. When DiskEdit is searching for your data you will see the following dialog box:



A screenshot of a teal dialog box titled "Searching for Data". It displays the progress of a search: "Wait |" followed by "Cylinder 1 Head 24 Sector 29". At the bottom, it says "ESC aborts".

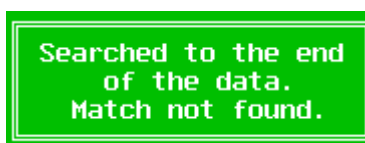
The cylinder, head and sectors are displayed as they are read. You may press the Escape key at any point to abort the search.



A screenshot of a green dialog box with the text "Search aborted" in white.

This dialog box will be shown if you pressed the Escape key.

You will see one of the following two dialog boxes if the search is allowed to continue to the end of the specified area.



You will see this dialog box if a match was not found. In this case, you will remain at the point from where you started the search.



This dialog box will appear if a match was found. The cursor will be moved to the area where the match was found, and the new sector will be displayed.

F5 - Continue search

The F5 key continues the search for the next occurrence of the data. If no search data has been entered, DiskEdit will prompt you to enter the data (as if you had pressed the F4 key).

F6 - Toggle edit mode

The F6 key toggles the current editing mode. The default editing mode is RAW. I.e. raw hexadecimal format. Depending on which portion of the disk you are editing, the F6 key toggles between RAW mode and a mode appropriate for that area.

For example, the F6 key toggles between RAW and Partition Table modes when you are editing the first sector of a physical disk.

The current editing mode can be manually selected using the F11 key.

Note:

Please consult Appendix I - DiskEdit Modules for a further description of the available edit modes.

F7 - Copy sectors to a file

The F7 key allows you to copy sectors to a file. You are prompted with a dialog box similar to one of the following three. This is for the FAT disk editing module:

```

Select logical sector

Cylinder: 0
          0 - 260 (0x104)
Head: 0
       0 - 254 (0xfe)
Sector: 1
        1 - 63 (0x3f)
LSN: N/A
      0 - 4192901 (0x3ffa85)
Cluster: N/A
         2 - 65506 (0xffe2)
Number: 1
        1 - 4192965 (0x3ffac5)
File:
      File to write disk sectors to

Enter accepts values
ESC aborts
```

This is for the RAW, HPFS and PIPE disk editing modules:

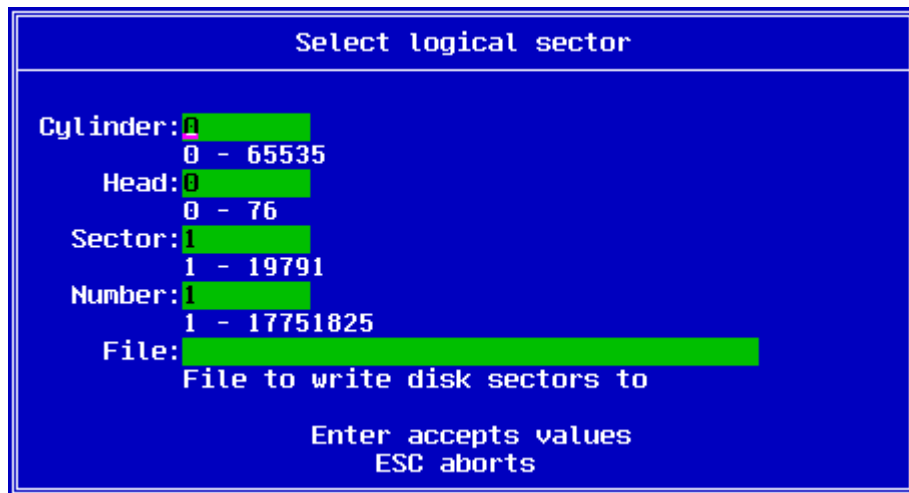
```

Select logical sector

Cylinder: 0
          0 - 565 (0x235)
Head: 0
       0 - 254 (0xfe)
Sector: 1
        1 - 63 (0x3f)
LSN: N/A
      0 - 9092726 (0x8abe76)
Number: 1
        1 - 9092790 (0x8abeb6)
File:
      File to write disk sectors to

Enter accepts values
ESC aborts
```

This is for the DEPHYS editing module:



You enter the starting cylinder, head and sector, the number of sectors as well as the file name which the data will be saved to. Once the data has been saved as a file, you may edit it by other means. The F8 key may be used to save the file back to the disk.

If no file name is entered, you will see this dialog box:



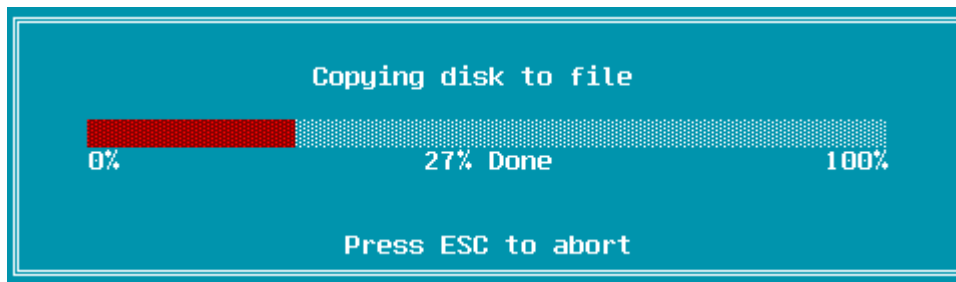
From there, you will be returned to the main editing window.

Note:

The file specification of the file refers to the local machine, not the remote machine, when using the remote disk editing option.

If you have multiple partitions and enough free disk space, you could perform an image backup of a complete disk by saving the entire disk to a file. That file could then be backed up to tape (or by other means). An image backup is a complete sector by sector save of the disk data. When you do an image restore, each sector is replaced to whence it originally came. Normal backups only save the file data, no information about the position of the files on the disk is saved. DiskEdit was not designed to do complete disk images, it was designed to do small amounts of disk information in this manner. DiskImg was written specifically to address the problems of imaging both logical and physical disks.

As the save continues, you will see a bar graph indicating the progress of the save.



F8 - Copy a file to sectors

The F8 key enables you to copy the data in a file directly to the disk. It is the complementary function to the F7 key. You will be prompted with a dialog box similar to one of the following three.

This is for the FAT disk editing module:

Select logical sector

Cylinder: 0
0 - 260 (0x104)

Head: 0
0 - 254 (0xfe)

Sector: 1
1 - 63 (0x3f)

LSN: N/A
0 - 4192901 (0x3ffa85)

Cluster: N/A
2 - 65506 (0xffe2)

Number: 1
1 - 4192965 (0x3ffac5)

File:
File to read disk sectors from

Enter accepts values
ESC aborts

This is for the RAW, HPFS and PIPE editing modules:

```

Select logical sector

Cylinder: 0
          0 - 565 (0x235)
Head: 0
       0 - 254 (0xfe)
Sector: 1
        1 - 63 (0x3f)
LSN: N/A
       0 - 9092726 (0x8abe76)
Number: 1
        1 - 9092790 (0x8abeb6)
File:
       File to read disk sectors from

Enter accepts values
ESC aborts

```

This is for the PHYS disk editing module:

```

Select logical sector

Cylinder: 0
          0 - 65535
Head: 0
       0 - 76
Sector: 1
        1 - 19791
Number: 1
        1 - 17751825
File:
       File to read disk sectors from

Enter accepts values
ESC aborts

```

You enter the starting cylinder, head and sector, the number of sectors as well as the file name which the data will be read from.

If no file name is entered, you will see this dialog box:

A file name was not specified.

From there, you will be returned to the main editing window.

Note:

The file specification of the file refers to the local machine, not the remote machine, when using the remote disk editing option.

Saving data directly to a position on the disk is a very dangerous operation. If you save data to the wrong portion of the disk the disk may become corrupted and totally unusable.

As a reminder of this, you are prompted with the following warning dialog box:

```
Warning! Warning! Warning! Warning! Warning!  
You are about to copy data from a file  
directly to the disk.  
This could cause unrecoverable damage  
to the data on the disk.  
DO YOU WISH TO CONTINUE?  
Press 'Y' to write the data.  
Any other key aborts the operation.
```

If you are certain that you wish to continue, press the Y key. Any other key aborts the operation, and the data will not be written to disk.

F10 - Exit

The F10 and Escape keys exit the DiskEdit program. You will be prompted to save any unsaved data if needed.

F11 - Manually select an editing mode

The F11 key enables you to manually select an editing mode. You only need to use this if the F6 key (Toggle editing mode) does not correctly identify the correct sector type. You will be prompted to save any unsaved data if needed.

Select a valid editing mode. Enter selects the current option. Escape aborts.							
PART TABL	-	Partition Table Editor	-	PHYS	RAW	HPFS	FAT PIPE
BOOT SECT	-	Boot Sector Editor	-		RAW	HPFS	FAT PIPE
FAT	-	FAT Editor	-				FAT
DIR	-	FAT Directory Editor	-				FAT
RAW	-	Raw Hexadecimal Editor	-	PHYS	RAW	HPFS	FAT PIPE
SUPER BLK	-	Super Block Editor	-			HPFS	
SPARE BLK	-	Spare Block Editor	-			HPFS	
BITMAP	-	Bitmap Editor	-			HPFS	
DIR BITMP	-	Directory Bitmap Editor	-			HPFS	
DIRBLK	-	Directory Editor	-			HPFS	
FNODE	-	FNode Editor	-			HPFS	
ALSEC	-	ALSec Editor	-			HPFS	
CODE PG I	-	Code Page Information Editor	-			HPFS	
CODE PG	-	Code Page Editor	-			HPFS	
BAD BLK L	-	Bad Block Editor	-			HPFS	
HOT FIX L	-	Hot Fix List Editor	-			HPFS	
BITMP LST	-	Bitmap List Editor	-			HPFS	

F12 - Jump to a sector

When editing a sector in a context sensitive mode, some of the fields are Logical Sector Numbers or Cluster Numbers. The F12 key allows you to jump directly to the sector specified. This enables you to easily navigate around the disk. If needed you will be prompted to save any unsaved data.

Examples

```
DISKEDIT A:
```

DiskEdit will edit drive A:. All other processes will be stopped from accessing drive A: whilst DiskEdit is using it.

```
DISKEDIT -p1
```

Edit the first physical hard disk. You will need this option to edit the partition table. To edit the drive safely, you will need to boot from a floppy, unless you have installed OS/2 on a second physical hard disk.

DISKEDIT C: -nolock

If any other processes, such as OS/2, are using drive C:, DiskEdit will not be able to lock it for exclusive access. Use the "-nolock" option to view the disk data. In this mode all editing functions are disabled. You are not allowed to change data, you are only allowed to view it.

DISKEDIT C: -on \\MAIN

DiskEdit will attempt to access C: on the remote machine named MAIN. All other disk editing features act as normal.

DiskImg - Disk Imaging Tool

Summary

DiskImg is a tool which allows you to image logical and physical disks. Disk to disk, disk to file and file to disk copies may be performed.

Icon



Command Line Format

Usage: DiskImg { Switches } <Drive Spec> <File Name>

Note:

<Drive Spec> is A: to Z: for logical disks

<Drive Spec> is 1: to 9: for physical disks

<File Name> is optional, if not specified the console is used

DiskImg can be called with no parameters. In this case, DiskImg goes into its interactive mode.

Switches

-r	Read from the disk, writing to the console.
-q	Quiet mode.
-w	Write to the disk, reading from the console.

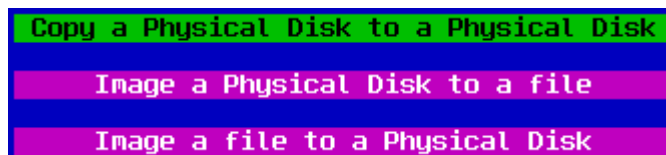
Description

DiskImg is a tool which enables you to image disks. A disk to disk copy can be performed, as can a disk to file as well as a file to disk copy. Both logical and physical disks can be handled. Although DiskImg can be totally command line driven though the use of switches, its default mode of operation is interactive, using a simple menu interface.

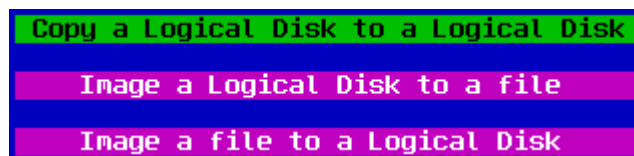
The initial menu with which you are presented is this:



You have a choice of either Physical (0:, 1: etc) or Logical (A:, B: etc) disk functions. Choose the required one, and you will be presented with either of these two menus:



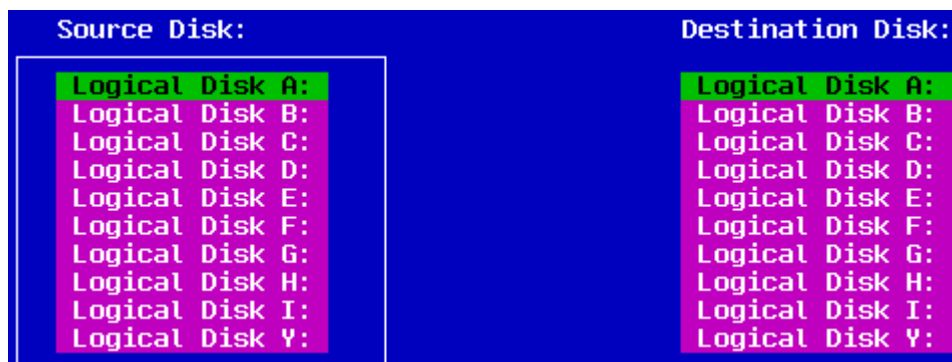
for the physical disk menu, and



for the logical disk menu.

Disk to disk imaging

For disk to disk copy to copy modes, you will be presented with a menu similar to this. The up and down arrow keys are used to select the appropriate drives. The tab key is used to switch between each set. Esc will abort, and the Enter key will accept the choices.



You will only see the above menu if there are multiple disks. If there is only one disk, you will see this:

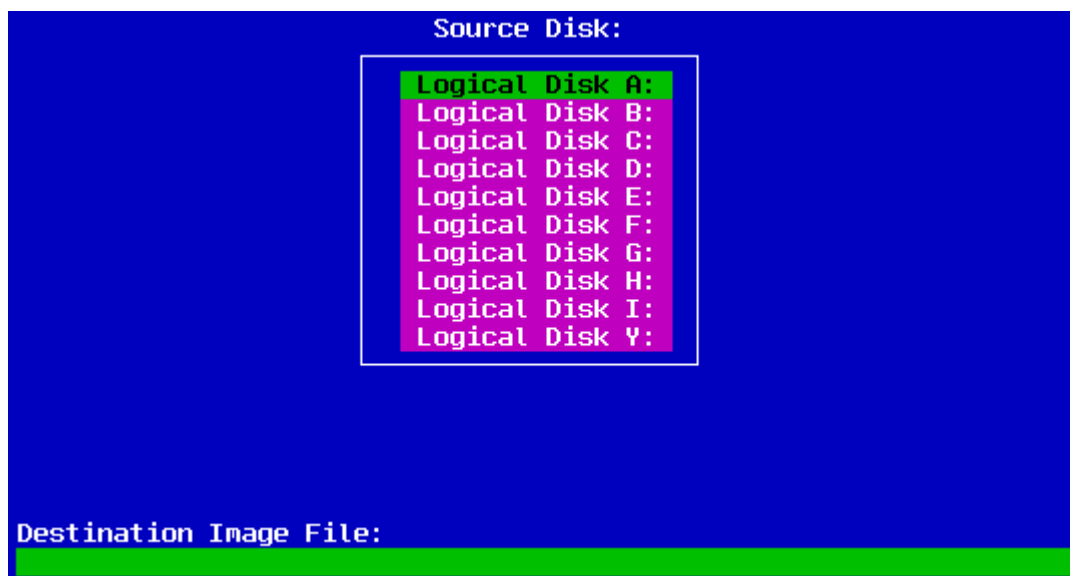


You will see a similar dialog box when there is only one logical disk.

The copy will proceed if both disks can be locked for exclusive access.

Disk to file imaging

If you wish to transport the disk, or delay the imaging process then you can use the Disk to file imaging modes. Here, you will be presented with a similar menu to the following:



Select the disk to image, and enter a filename. The disk will be imaged to that file. If no filename is entered, you will be reminded with this dialog box:

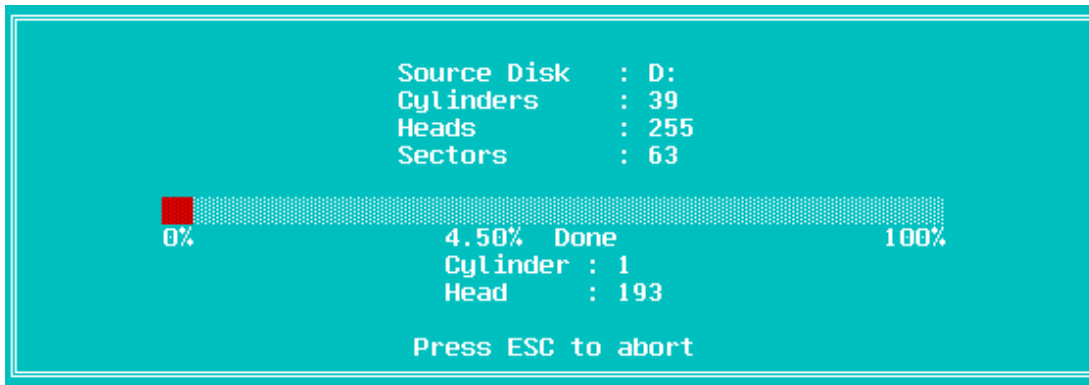


If the file already exists, you will be prompted with this dialog box:



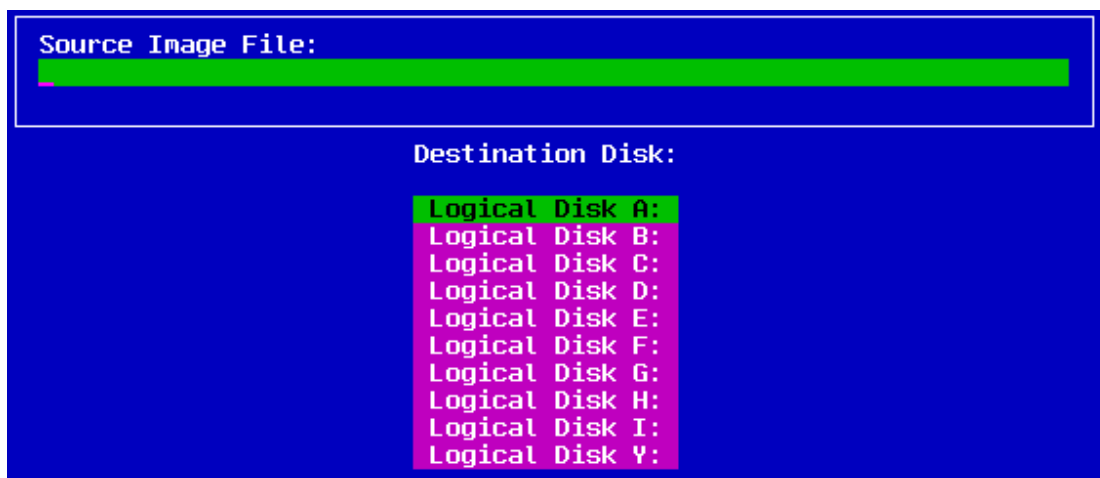
Press the Y key to overwrite the existing file, or any other key to abort the process.

Once all of these tests have been passed, and the disk is locked, the copy can proceed. As it proceeds, you will be informed as to its progress via this dialog box:



File to disk imaging

To create a previously saved image, the file to disk modes can be used. In this instance, you will be presented with a dialog box similar to this:



Enter the appropriate values and press enter. If everything is correct, the copy will proceed as described above.

-r Read from the disk

This switch forces DiskImag to read from the specified disk (specified on the command line) and the output is written to the console. This allows the output to be redirected to a file or piped to another process.

-q Quiet mode

This switch stops DiskImag from displaying any windows or other screen output.

-w Write to the disk

This switch forces DiskImag to read its input from the console and write to the specified disk (specified on the command line). This allows a file to be redirected into, or piped from another process into DiskImag.

F1 - Help

Whilst DiskImg displays "Press F1 for help" in the lower right hand corner of the window, help is available. Depending on which menu you are displaying, it will look similar to this:



Examples

DiskImg

DiskImg will be started in its default interactive mode.

```
DiskImg -r C: > OUT
```

DiskImg will read drive C: and it will be redirected to the file OUT.

```
DiskImg -r F: | gzip > drive-f.gz
```

Drive F: will be read and the output will be piped into the GZIP program, which will compress the data into the file DRIVE-F.GZ.

```
gunzip < drive-f.gz | DiskImg -w F:
```

Gunzip will decompress the file drive-f.gz on the fly (ie without creating an uncompressed intermediate file) and DiskImg will write it to drive F:

EABackup - Backup Extended Attributes

Summary

EABackup backs up all of the Extended Attributes of matching files into one archive file.

Icon



Command Line Format

Usage: EABackup {Switches} <Archive File> <File Spec>
<Drive Spec(s)>

Switches

-a	All disks.
-d	Delete EA's after archive.
-h	Hard disks only.
-s	Include subdirectories in the search.

Note:

If no drive specifications are entered the current drive is the only one searched.

Description

EABackup is a utility which backs up the Extended Attributes of files which match a given file specification into a Extended Attribute Archive file. EABackup can optionally delete the Extended Attributes which are attached to the files after they have been archived.

EABackup is a super set of the standard OS/2 program EAUTIL. EAUTIL's biggest limitation is that it can not deal with wild cards. EABackup overcomes this problem.

EABackup's biggest use is to save the Extended Attributes of files when they are going to be operated upon by any non Extended Attribute aware program.

This includes all DOS programs and even some OS/2 programs. This is mostly used on FAT partitions which are going to be defragmented using a DOS program. EABackup will also save the extended attributes of files on HPFS partitions.

EARestor is used to restore the Extended Attributes of the files in the archive.

-a All disks

EABackup will scan all logical disk drives defined by the system, starting at drive A:. The drives are scanned in order, starting at A:, then B: and so on.

-d Delete Extended Attributes after archive

EABackup will delete the Extended Attributes of each file after they have successfully archived.

-h Hard disks only

EABackup will scan all logical disk drives defined by the system, starting at drive C:. The drives are scanned in order, starting at C:, then D: and so on.

-s Subdirectories

EABackup will include subdirectories as part of its search of the specified disk drives. If a subdirectory name is included as part of the file specification, then the search will include that subdirectory and all subdirectories below it. If no subdirectory is given as part of the file specification, then the current directory and all below it are included in the search.

Examples

```
EABackup ALLFILES.DAT -h
```

EABackup will archive all Extended Attributes of all files on all drives into the file ALLFILES.DAT.

```
EABackup Drive-C.EA C:\ /sd
```

EABackup will save the Extended Attributes of all files on drive C:, deleting them as they are archived into the file DRIVE-C.EA.

EABind - Automatically bind EA's to files

Summary

EABind binds data to files as extended attributes.

Icon



Command Line Format

Usage: EABind {Switches} <File>

OR:

Usage: EABind {Switches} <File> <Type> <EA Name> <EA Data>

Switches

- n Set EA as needed.
- s Include subdirectories in the search.

Description

EABind binds data to files as extended attributes. For a discussion on Extended Attributes see Appendix A. EABind needs to know the file to bind the data to, the type of the extended attribute, the extended attribute name and the data to be bound. EABind can process a response file or it can accept parameters from the command line. The response file contains the file name, the extended attribute type, the extended attribute name and the extended attribute data.

With version 2, EABind has now been expanded to include wild card processing. The <File> parameter can be given a wild card specification. The wild card specification can be provided on either the command line or in a response file.

The EA Data field varies depending on the EA type.

Type	Definition of EA Data
EAT_ASCII	The ASCII string to be bound.
EAT_BINARY	The file name of the binary data to be bound.
EAT_BITMAP	The file name of the bitmap to be bound.
EAT_ICON	The file name of the icon to be bound.
EAT_METAFILE	The file name of the metafile to be bound.

-n Set EA as needed

Extended attributes (EAs) are non-critical by default. A non-critical EA is one which is not necessary to the functionality of the application. If a non-critical EA is lost, the system continues to operate correctly. For example, losing the icons associated with data files does not generally cause any ill effect other than the inability to show the icon.

A critical extended attribute is one which is necessary for the correct operation of the operating system or of a particular operation. EAs should be marked as critical if their loss would cause the system or program to perform incorrectly. For example, a mail program might store mail headers in EAs. The loss of the header from a message would normally render the mail program unable to further use that message. This would be unacceptable, so the mail program should mark this EA as critical.

This switch sets the EA as critical or needed.

-s Subdirectories

EABind will include subdirectories as part of its search for files. The current subdirectory is searched as well as all subdirectories below the current one.

Examples

```
EABind bindfile
```

Bindfile is the sample file provided for EABind. It sets the .COMMENTS EA to the default values for the files provided in the Graham Utilities. It also sets the .ICON EA for the files so that they will appear as icons on the desktop.

```
EABind C:\STARTUP.CMD EAT_ASCII .COMMENTS "This is the system
startup file"
```

EABind will bind the text string "This is the system startup file" to C:\STARTUP.CMD. The EA name will be .COMMENTS and the type will be EAT_ASCII. This will overwrite any previous .COMMENTS extended attribute. Any other extended attribute will be left unchanged.

```
EABind *.Ico EAT_ASCII .TYPE Icon
```

This will get EABind to bind the text string "Icon" to all .ICO files in the current directory. The EA name will be .TYPE and the type will be EAT_ASCII.

```
EABind *.Ico EAT_ASCII .TYPE Icon -s
```

EABind will bind the text string "Icon" to all .ICO files in the current subdirectory and all subdirectories below it. The EA name will be .TYPE and the EA type is EAT_ASCII.

EADump - Dump Extended Attributes of a file

Summary

EADump will perform a hexadecimal dump of all extended attributes attached to a file.

Icon



Command Line Format

Usage: EADump {Switches} <File Spec> <Drive Spec(s)>

Switches

- a All disks.
- h Hard disks only.
- l List the extended attribute names only.
- n<Name> Dump extended attributes of a given name.
- o Only list files with extended attributes.
- r Dump EAT_MVMT and EAT_MVST data as raw hex.
- s Include subdirectories in the search.

Description

EADump will perform a hexadecimal dump of all extended attributes attached to a particular file. The extended attribute name, type and flag value are listed followed by the hexadecimal dump of the extended data itself.

A simple example of output is:

```
Searching Drive E:
Dumping: E:\os2utils\be.ico
EA Name: .TYPE
EA Type: Length preceded ASCII
EA Flag: 00
000000 fd ff 04 00 49 63 6f 6e          ŷ...Icon
8 bytes dumped
```

Extended Attribute Data Format

The data which is displayed consists of more than just the extended attribute data itself. It also includes the extended attribute name and type of each extended attribute.

The first word (2 bytes) of the extended attribute data represents the extended attribute data type. The second word of the data is the length of the data itself, not including the first four bytes.

Intel Byte Ordering

Intel processors use a low byte-high byte order to represent whole numbers in word (2 byte) format. This means if you wish to extract a word value when you read the data as a series of bytes, then you must reverse the two bytes. Therefore: 0123 in hexadecimal (or 291 decimal) would be read as 23 01 as a series of bytes.

EAT_ Values

The standard extended attribute values are defined as:

EA Type	Value	Description
EAT_BINARY	0xFFFE	length preceded binary
EAT_ASCII	0xFFFD	length preceded ASCII
EAT_BITMAP	0xFFFB	length preceded bitmap
EAT_METAFILE	0xFFFA	length preceded metafile
EAT_ICON	0xFFF9	length preceded icon
EAT_EA	0xFFEE	length preceded ASCII name of associated data
EAT_MVMT	0xFFDF	multi-valued, multi-typed field
EAT_MVST	0xFFDE	multi-valued, single-typed field
EAT_ASN1	0xFFDD	ASN.1 field

Values 0xFFFE through 0x8000 are reserved for use by OS/2.

Values 0x0000 through 0x7FFF are user definable.

Value 0xFFFC is not used.

Therefore, in the above example the FD FF is read as a word of FFFD, hence the type of EAT_ASCII. The length 04 00 reads as 0004 or 4 bytes of actual data.

EA Flag Values

The EA flag value specifies one or more flags. In OS/2 version 1.2 (and above), the only flag available is FEA_NEEDED, indicating an extended-attribute bit is needed. FEA_NEEDED is defined as 80 hex. The EA flag can be either zero or FEA_NEEDED.

-a All disks

EADump will scan all logical disk drives defined by the system, starting at drive A:. The drives are scanned in order, starting at A:, then B: and so on.

-h Hard disks only

EADump will scan all logical disk drives defined by the system, starting at drive C:. The drives are scanned in order, starting at C:, then D: and so on.

-l List the extended attribute names only

EADump will not perform a hexdump of the extended attribute data. It will only display the extended attribute name, type and flag values. It will display the file name (even if the file does not have any extended attribute data attached to it) unless the -o switch was also specified.

-n Dump extended attributes of a given name.

By default, EADump will dump the data of every extended attribute attached to a file. Using this switch you may specify one particular extended attribute name. Only the extended attribute data of files which have an extended attribute of the same name will be dumped. The case of the name is not important.

-o Only list files with extended attributes

EADump will only display the details of files which have extended attributes attached to them. It will perform a hexdump of the data - unless the -l switch was also specified.

-r Dump EAT_MVMT and EAT_MVST data as raw hex

EADump's default behaviour is to attempt to dump EAT_MVMT and EAT_MVST data types in their component form. This switch forces EADump to dump the EA's of these types as plain raw data. No type interpretation is attempted.

-s Subdirectories

EADump will include subdirectories as part of its search of the specified disk drives. If a subdirectory name is included as part of the file specification, then the search will include that subdirectory and all subdirectories below it. If no subdirectory is given as part of the file specification, then the current directory and all below it are included in the search.

Examples

```
EADUMP *.exe
```

EADump will dump the extended attributes of any .exe files which it finds in the current directory.

```
EADUMP C:\OS2\*.* -ols
```

EADump will list the file names, extended attribute names, types and flag values of all files in C:\OS2 and its subdirectories.

```
EADUMP C:\STARTUP.CMD
```

EADump will dump the extended attributes of C:\STARTUP.CMD only.

```
EADUMP
```

EADump will dump the extended attributes for all of the files in the current directory.

```
EADUMP -hol
```

EADump will list the file name, extended attribute name, type and flag values of all files with extended attributes across all hard disks.

```
EADUMP -n.Type *.Ico
```

EADump will dump the extended attributes of all .ICO files in the current directory that have an extended attribute named ".TYPE".

The OS/2 System Editor (E.EXE) attaches a EAT_MVMT EA to files that it edits.

```
EADUMP LMAX.CMD
```

EADump will produce output similar to this:

```
EA Name: .TYPE
EA Flag: 00
EA Type: Multi-valued, multi-typed field
0000 df ff 00 00 01 00          ß.....
6 bytes dumped

EA Type: Length preceded ASCII
0000 fd ff 0a 00 50 6c 61 69 6e 20 54 65 78 74  ý...Plain
Text
14 bytes dumped
```

```
EADUMP -r LMAX.CMD
```

EADump will produce output similar to this:

```
EA Name: .TYPE
EA Flag: 00
EA Type: Multi-valued, multi-typed field
0000 df ff 00 00 01 00 fd ff 0a 00 50 6c 61 69 6e 20
...ý...Plain
0010 54 65 78 74 00
Text.
21 bytes dumped
```


EARestor - Restore Extended Attributes

Summary

EARestor restores the Extended Attributes of files which were archived using EABackup.

Icon



Command Line Format

Usage: `EARestor <Archive File>`

Description

EARestor is used to restore the Extended Attributes back to the files as they were archived in the archive file. The complete path of the original files is stored in the archive file, so that one archive file can save the Extended Attributes of multiple disks.

Example

```
EARestor ALLFILES.DAT
```

EARestore will restore all of the Extended Attributes which were saved by EABackup in the file ALLFILES.DAT.

FAT-Bad - Set bad clusters on FAT disks

Summary

FAT-Bad allows you to mark clusters as bad on FAT partitions.

Icon



Command Line Format

Usage: FAT-Bad {Switches} Drive Spec <N1> [N2]

Switches

-on Run on <Machine Name>.

Parameters

- N1 This is the first cluster to be marked as bad. If it is the only parameter given, then only this cluster is marked bad. This parameter must be supplied.
- N2 This is the last cluster to be marked as bad. This is an optional parameter. If it is given, then the first to last cluster (inclusive) will be marked bad.

Note:

Both of the parameters can be given in either decimal or hexadecimal. Decimal is the default. If you wish to enter values in hexadecimal, simply prefix the numbers with a "0x".

Note:

The value of the parameters are in Clusters. They start at two (2) and go up to the maximum as defined by FAT-Info in the Number of clusters field.

Description

FAT-Bad allows you to mark clusters on FAT disks as bad, so that they will not be used.

The clusters to be marked as bad, must first be free for allocation. If they are already in use, FAT-Bad will report it as such, and the sector will not be marked as bad. FAT-View may be used to determine if the sector is in use, and if so, by what.

-on Run on <Machine Name>

This switch forces DEPIPE to be loaded and allows FAT-Bad to access disk drives located on remote machines. The <Machine Name> must be specified with a leading "\\\". The machine name of the remote machine will have been specified when the network software on the remote machine was installed.

Note:

DESVR must be running on the remote machine for this option to work.

Consult Appendix H - DESVR for further details.

Examples

```
FAT-Bad
```

This will report an error as no drive specification or starting sector number was given.

```
FAT-Bad E: 1000
```

FAT-Bad will attempt to mark cluster 1000 (decimal) as bad.

```
FAT-Bad E: 0x1000
```

FAT-Bad will attempt to mark cluster 1000 hexadecimal (4096 decimal) as bad.

```
FAT-Bad F: 1000 0x2000
```

FAT-Bad will attempt to mark clusters 1000 to 8192 (decimal), inclusive as bad.

FAT-Dfrg - Defragment FAT files

Summary

FAT-Dfrg defragments specified files on HPFS volumes.

Icon



Command Line Format

Usage: FAT-Dfrg {Switches} <File Spec> <Drive Spec(s)>

Switches

-h	Hard disks only.
-o<n>	Optimise files with greater than <n> extents.
-s	Subdirectories.
-on	Run on <Machine Name>.

Description

FAT-Dfrg is a command line utility which simply reports the number of extents (or fragments) that each file matching the given file specification is in, and then defragments them if they are in more than a specified amount of fragments.

FAT-Dfrg is capable of defragmenting one file, one directory, one drive or all drives in one invocation.

If you tell FAT-Dfrg to defragment a non FAT drive you will get the following message:

```
Drive is not FAT.
```

As FAT-Dfrg runs, it searches all specified files and determines the number of fragments each file is in. If it is in a greater number of extents than specified (a default of two), defragmentation of that file is attempted. For a file to be successfully defragmented there must be enough free space on the drive for a

copy of the file. If there is insufficient space on the drive to defragment a particular file you will see the message:

```
<filename> Comprises <n> extents (<x> clusters). Unable to find <x> free contiguous clusters.
```

FAT-Dfrg must also be able to obtain exclusive access to the disk. If any other processes are accessing the disk, FAT-Dfrg will not defragment the file and you will be prompted with this message:

```
SYS0108: The disk is in use or locked by another process.
```

Note:

FAT-Dfrg attempts to defragment a file by simply copying it to an area on the disk which is of equal size. There is no regard as to the new position of the file. This may actually fragment your free space further than it is already. Also, directories are not moved in this operation. For a complete, true packing of your directories and files, please see FAT-View (with the -d switch).

FAT-Dfrg does not attempt to defragment the file into two or more extents. It will attempt to defragment the file into one extent only.

-h Hard disks only

FAT-Dfrg will scan all logical disk drives defined by the system, starting at drive C:. The drives are scanned in order, starting at C:, then D: and so on.

-o Specify optimisation threshold

By default, FAT-Dfrg attempts to defragment files with more than two extents. You may use this switch to change the default. However, if you set the optimisation threshold to too small a number you may fragment the free spaces. You can use FAT-View and FAT-Info to check the number of free spaces.

-s Subdirectories

FAT-Dfrg will include subdirectories as part of its search of the specified disk drives. If a subdirectory name is included as part of the file specification, then the search will include that subdirectory and all subdirectories below it. If no subdirectory is given as part of the file specification, then the current directory and all below it are included in the search.

-on Run on <Machine Name>

This switch forces DEPIPE to be loaded and allows FAT-Dfrg to access disk drives located on remote machines. The <Machine Name> must be specified with a leading "\\\". The machine name of the remote machine will have been specified when the network software on the remote machine was installed.

Note:

DESVR must be running on the remote machine for this option to work.

Consult Appendix H - DESVR for further details.

Examples

FAT-Dfrg

FAT-Dfrg will attempt to defragment all files in the current directory.

FAT-Dfrg /s

FAT-Dfrg will attempt to defragment all files in the current directory and all directories below it.

FAT-Dfrg -h

FAT-Dfrg will attempt to defragment all files with more than two extents on all hard disks.

FAT-Dfrg -o3

This changes the optimisation threshold to files with more than three extents and attempts to optimise all files in the current directory.

FAT-DFRG -on \\DATA C:\ /s

FAT-Dfrg will attempt to defragment all files with more than two extents on drive C: on the remote machine named DATA.

FAT-Ext - Report on FAT file fragmentation

Summary

FAT-Ext reports on the number of extents (fragments) of files.

Icon



Command Line Format

Usage: FAT-Ext {Switches} <File Spec> <Drive Spec(s)>

Switches

-e	Extents first
-h	Hard disks only.
-s	Subdirectories.
-on	Run on <Machine Name>.

Description

FAT-Ext is a command line utility which reports the number of extents (or fragments) which each file matching the given file specification is in.

FAT-Ext is capable of reporting on one file, one directory, one drive or all drives in one invocation. This can be particularly useful if you wish to check to see if the drive needs defragmentation. The output of FAT-Ext may be redirected to either a file or to a printer to save the output.

If you ask FAT-Ext to report on a non FAT drive you will get the following message:

```
Drive is not FAT.
```

-e Extents first

The default behaviour of FAT-Ext is to list the file name followed by the number of extents of which each file comprises. The -e switch forces FAT-Ext to list the extents first. This allows the output of FAT-Ext to be piped into smart sort filters (such as GNU's GSORT) which can sort in numeric order. This will give you a list of files in fragmentation order, not file found order.

-h Hard disks only

FAT-Ext will scan all logical disk drives defined by the system, starting at drive C:. The drives are scanned in order, starting at C:, then D: and so on.

-s Subdirectories

FAT-Ext will include subdirectories as part of its search of the specified disk drives. If a subdirectory name is included as part of the file specification, then the search will include that subdirectory and all subdirectories below it. If no subdirectory is given as part of the file specification, then the current directory and all below it are included in the search.

-on Run on <Machine Name>

This switch forces DEPIPE to be loaded and allows FAT-Ext to access disk drives located on remote machines. The <Machine Name> must be specified with a leading "\\\". The machine name of the remote machine will have been specified when the network software on the remote machine was installed.

Note:

DESVR must be running on the remote machine for this option to work.

Consult Appendix H - DESVR for further details.

Examples

FAT-Ext

FAT-Ext will report the number of extents of all files in the current directory.

FAT-Ext /s

FAT-Ext will report the number of extents of all files in the current directory and all directories below it.

FAT-Ext -h

FAT-Ext will scan all hard disks.

FAT-Ext C:\OS2

List the number of extents of all files in C:\OS2.

FAT-Ext C:\OS2 /s

FAT-Ext will list the number of extents of all files in C:\OS2 and all directories below it.

FAT-Ext E:\Data -on \\MAIN

FAT-Ext will report the number of extents of all files in E:\DATA on the remote machine named MAIN.

FAT-Info - Report on FAT internals

Summary

FAT-Info provides a quick overview of the structure and layout of your FAT disk.

Icon



Command Line Format

Usage: FAT-Info {Switches} <File Spec> <Drive Spec(s)>

Switches

- b Display bad sector lists.
- f Display the free spaces.
- h Hard disks only.
- on Run on <Machine Name>.

Description

FAT-Info provides a brief list of information pertaining to the layout and use of your FAT disk.

The information which it returns is:

FAT Type	16 bit or 12 bit. This is the number of bits used to encode each entry in the FAT.
1st FAT starts at	This is the LSN of the first sector of the first copy of the FAT.
2nd FAT starts at	This is the LSN of the first sector of the second copy of the FAT.

Root Directory starts at	This is the LSN of the first sector of the Root Directory.
Root Directory length	This is the number of sectors which the Root Directory occupies.
Data area starts at	This is the LSN of the first sector of the data area of the disk. This is also the first sector of cluster two.
Number of clusters	This is the total number of clusters on the disk. The numbers in the brackets are the first and last numbered clusters.
Number of free clusters	This is the current number of free clusters available on the disk.
Number of free spaces	This is the current number of free spaces on the disk. Ideally, there should be one.
Cluster size (bytes)	This is the number of bytes which each cluster occupies. It will be between 512 and 32,768.

When the information returned is a logical sector number, it is also displayed as a Cylinder, Head and Sector.

If you ask FAT-Info to report on a non FAT drive you will get the following message:

```
Drive is not FAT.
```

-b Display bad sector lists

This switch displays the clusters which have been marked as bad.

-f Display the free spaces

The -f switch forces to FAT-Info to display the free space of the disk. The cluster numbers of free clusters are displayed.

-h Hard disks only

FAT-Info will scan all logical disk drives defined by the system, starting at drive C:. The drives are scanned in order, starting at C:, then D: and so on.

-on Run on <Machine Name>

This switch forces DEPIPE to be loaded and allows FAT-Info to access disk drives located on remote machines. The <Machine Name> must be specified with a leading "\\\". The machine name of the remote machine will have been specified when the network software on the remote machine was installed.

Note:

DESVR must be running on the remote machine for this option to work.

Consult Appendix H - DESVR for further details.

Examples

FAT-Info

This will produce the minimum amount of information which FAT-Info can extract from the file system.

FAT-Info /b

FAT-Info will display all of its normal information. In addition it will list all sectors which are currently marked as bad.

FAT-Info -bfh

FAT-Info will scan all hard disks and report on the maximum amount of information available.

FAT-Info C: -bfm -on \\WARP

FAT-INFO will display all of the information available about C: on the remote machine named WARP.

FAT-Null - Wipe FAT disks

Summary

FAT-Null wipes FAT disks on a cluster by cluster basis.

Icon



Command Line Format

Usage: FAT-Null {Switches} Drive Spec(s)

Switches

- g Government security rules for wiping.
- r Repeat wipe; default of 1.
- v wipe Value; default of 0.
- on Run on <Machine Name>.

Description

FAT-Null wipes all free clusters on a HPFS disk. Sectors which are occupied by files are not wiped. For FAT-Null to be able to process the disk, it must be able to lock the disk. If the disk is in use by another process, FAT-Null will inform you that it is unable to lock the disk and exit. FAT-Null must lock the disk, so that it can be assured that the information which it is reading is accurate. FAT-Null displays the cluster number of each cluster as it is being wiped. The escape key may be pressed at any point in time to abort the wiping process.

In order to help protect you from accidental erasures you must enter a drive specification.

If FAT-Null is asked to operate on a non FAT drive you will get the following message:

```
Drive is not FAT.
```

-g Government security rules for wiping

This switch forces FAT-Null to erase the file to the US Department of Defence security specification DoD 5220.22-M. It writes over all of the data area of the file with 1's (FFh) then writes over it all again with 0's (00h). By default, it repeats this procedure 3 times. The optional <n> parameter for this test overrides the default repeat count of 3. Finally, the data area of the file is overwritten with the value of F6h.

-r Repeat wipe count

The default wipe count value is 1. This switch specifies a new repeat count. The repeat count is the number of times which the disk is overwritten.

-v Wipe Value

The default value of the data which is written to the disk is zero (0). Use this switch to specify another value.

-on Run on <Machine Name>

This switch forces DEPIPE to be loaded and allows FAT-Null to access disk drives located on remote machines. The <Machine Name> must be specified with a leading "\\\". The machine name of the remote machine will have been specified when the network software on the remote machine was installed.

Note:

DESVR must be running on the remote machine for this option to work.

Consult Appendix H - DESVR for further details.

Example

```
FAT-Null F:
```

```
Run FAT-Null on drive F:
```

FAT-UD - Undelete files on FAT drives

Summary

FAT-UD allows you to undelete files located on FAT drives.

Icon



Command Line Format

Usage: FAT-UD {Switches} <Drive Spec(s)>

Switches

-l	Lock the drive.
-p	Prompt for restore path.
-s	Subdirectories are created.
-on	Run on <Machine Name>.

Description

FAT-UD scans the disk for deleted directory entries, validates them and then builds up a list to enable you to select the files you wish to recover.

When FAT-UD starts up you will see the following dialog box:



When FAT-UD has finished scanning the disk you will either see a dialog box informing you that there were no deleted files found, or a window similar to:

```

FAT UnDelete
Deleted: 38      Tagged: 0      Drive: A:  Prompt: Off  Subdirs: Off
152 / 152      1,239,234  \TAPCIS\DOWNLOAD\?TS304.ZIP
1002 / 1002    8,205,323  \TAPCIS\DOWNLOAD\?TU302.ZIP
  1 /  1              51  \WINWORD2\GU-GEN\?$HARSET.DOC
  3 /  3              23,463 \WINWORD2\GU-GEN\?DOC0017.TMP
 11 / 11              82,100 \WINWORD2\GU-GEN\?DOC0004.TMP
  1 /  1              51  \WINWORD2\ARTICLES\?$ANSLAT.DOC
  1 /  1              51  \WINWORD2\?$ARPFAX.DOT
  1 /  1              779  \HTDOGPRO\PUBLISH\?ABLE.BAK
  4 /  4              24,836 \?CSI\?FDISK.CWA
 21 / 21          165,351 \?CSI\?FDISK.EXE
  2 /  2              9,022  \?CSI\?FDISK.HLP
  4 /  4              28,776 \?CSI\?SPI2DOS.SYS
  2 /  2              14,314 \?CSI\?SPI4DOS.SYS
  5 /  5              36,160 \?CSI\?SPI7DOS.SYS
  4 /  4              29,968 \?CSI\?SPI8DOS.SYS
  1 /  1              4,608  \?CSI\?SPIBUF.SYS
  4 /  4              29,368 \?CSI\?SPICD.SYS
  2 /  2              13,538 \?CSI\?SPIDISK.SYS
  2 /  2              10,720 \?CSI\?SPIEDOS.SYS
  7 /  7              50,542 \?CSI\?DPLAYER.EXE
  7 /  7              50,880 \?CSI\?DPLAYER.FON
  2 /  2              10,090 \?CSI\?A160.SYS
Press F1 for Help

```

FAT-UD allows you to select multiple files for recovery in one operation. The Enter key undeletes the currently selected file and Control-Enter allows you to undelete all tagged files. An individual file is tagged using the space bar.

When a file is deleted under the FAT file system, the space is freed for use and the first character of the file name is changed. The changed character is replaced by a '?'. You will have to edit this to recover the file, as '?' is not a valid character.

The two numbers to the left of the screen represent the number of clusters available for recovery and the calculated number of clusters which the file should contain. This will give you an idea as to the likelihood of a successful recovery.

If you have turned prompting on, or the file to restore already exists, or the filename contains invalid characters, you will be prompted to edit it with a dialog box similar to the following:

```

Edit the path
The file name contains invalid characters. Please edit it:
H:\?CSI\?FDISK.CWA

```

As the files are being recovered you will be informed as to which file is currently being recovered.



-l Lock the drive

This switch attempts to lock the drive so that other processes can not access the drive whilst FAT-UD is accessing the drive. This is to ensure that no other processes inadvertently overwrite the files which you are trying to recover.

-p Prompt for restore path

By default, the files of the given path are automatically written to the disk. This switch turns on prompting. You will be prompted to edit the complete path and filename of the files as they are being restored.

-s Subdirectories are created

By default, all path information is removed from the filename. This switch turns on subdirectory processing. It leaves the subdirectories intact.

-on Run on <Machine Name>

This switch forces DEPIPE to be loaded and allows FAT-UD to access disk drives located on remote machines. The <Machine Name> must be specified with a leading "\\\". The machine name of the remote machine will have been specified when the network software on the remote machine was installed.

Note:

DESVR must be running on the remote machine for this option to work.

Consult Appendix H - DESVR for further details.

Whilst FAT-UD is running, the function keys listed below have the following functions associated with them.

F1 - Help

The help for FAT-UD is accessed by pressing the F1 key. If the F1 key is pressed you will see the following dialog box:

```
FAT UnDelete
The Graham Utilities for OS/2 V2.00.

F1   This help screen
F2   Sort the file order
F3   Select restore disk
F4   Tag all files
F5   Toggle all files
F6   Toggle Prompt
F7   Toggle Subdirectories
F8   Search and Tag files
F9   Search and Replace files
F10  Enter initial restore path
Esc  Exit FAT-UD
Space Tag/Untag a file
←   Undelete current file
^←  Undelete all tagged files

<Press any key to continue>
```

F2 - Sort the file order

By default, FAT-UD displays the files in the order in which they are found. The F2 key instructs FAT-UD to sort the file list in alphabetical order. As the sort progresses you will see this:

```
Sorting...
25
```

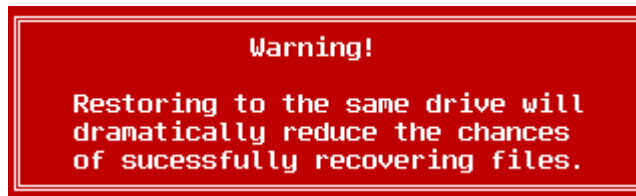
The number which is displayed is the number of entries which have been sorted so far.

F3 - Select restore disk

A: is the default drive to which the deleted files are to be restored. If you wish to change the drive, press the F3 key and you will be presented with the select restore drive dialog box:



Use the arrow keys or simply press the required drive letter. If you choose the same drive as that from which you are restoring, you will be warned with the following dialog box:



This is because you are likely to overwrite the file that you are trying to recover!

F4 - Tag all files

The F4 key tags all files which have been found.

F5 - Toggle all files

The F5 key toggles the tagged state of all files.

F6 - Toggle prompt

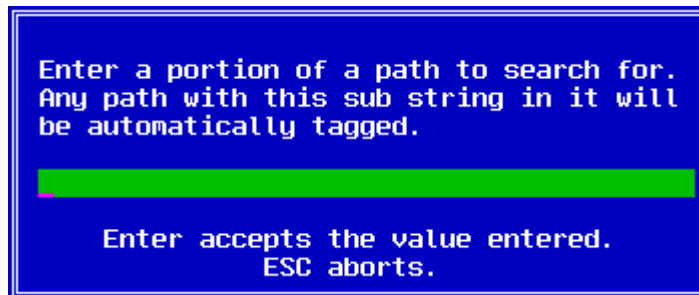
The F6 key toggles the current state of prompting. When prompting is on you will be asked to edit the path and filename of all restored files.

F7 - Toggle subdirectories

The F7 key toggles the current subdirectory state. When subdirectories are on, the path information from the files is not removed. With subdirectories off, all path information is removed and all files are restored into the root directory of the restore drive. All subdirectories are automatically created for you.

F8 - Search and tag files

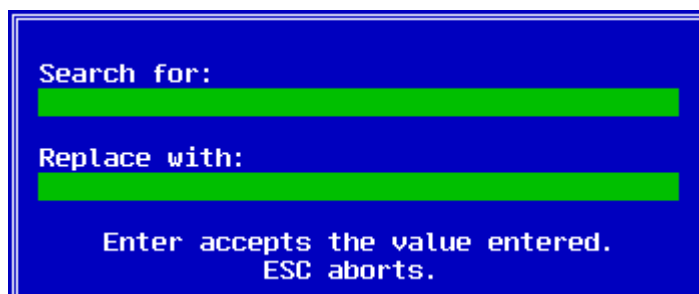
The F8 key allows you to perform an automatic search and tag function on all of the files which have been found.



You may enter a sub string which is compared to all of the file names. If there is a match the file is tagged. This is useful if you wish to recover all files from a particular directory.

F9 - Search and replace files

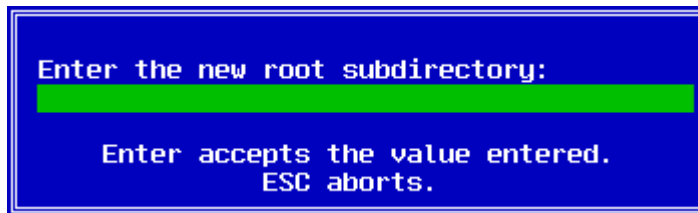
The F9 key allows you to change the file name and path of the files which have been found. This enables easy editing of multiple files all at once. It is particularly useful if you have multiple files all from the same deleted directory. You can pre-edit the directory path name, thus saving you some work later on at file recovery time.



Enter the characters which you wish to search for and the characters to replace with and press Enter.

F10 - Enter initial restore path

The F10 key allows you to specify an initial restore path off the root directory. This option allows you to graft the restored files to a new path. By default, the files are restored to the paths as they appear. This may cause restored files to be mixed up with normal files. There are circumstances where you may wish to separate the restored files for further examination.



For example, a file "`\MAX\FILE\WARPSPEED\GU-DEMO.LZH`" would normally be restored to "`\MAX\FILE\WARPSPEED\GU-DEMO.LZH`". If you were to specify a new restore path of "RECOVERED", then using the above example, the restored file would be "`\RECOVERED\MAX\FILE\WARPSPEED\GU-DEMO.LZH`".

Esc - Exit

The escape key exits FAT-UD.

Examples

```
FAT-UD
```

FAT-UD will scan for deleted files on the current drive.

```
FAT-UD /s
```

FAT-UD will scan for deleted files on the current drive and subdirectory processing will initially be turned on.

FAT-UD H: -d

FAT-UD will scan drive H: for valid files. This option will work on both valid and damaged volumes.

FAT-UD D: -l

FAT-UD will attempt to lock drive D: and search for deleted files. Using this option you will not be able to restore files to the same drive. If FAT-UD can not lock the drive it will not continue.

FAT-UD F: -on \\BBS /s

This scans for deleted files on drive F: on the remote machine named BBS. Subdirectory processing will initially be turned on.

FAT-View - View FAT drives

Summary

FAT-View allows you to view the contents and layout of your FAT volumes. As an option it can also visually defragment them.

Icon



Command Line Format

Usage: FAT-View {Switches} <Drive Spec(s)>

Switches

- c Clear the free clusters after defragment.
- d Defragment files.
- g Government security rules for wiping.
- n Number of buffers; default of 100.
- r Repeat wipe count; default of 1.
- u Unattended mode.
- v wipe Value; default of 0.
- on Run on <Machine Name>.

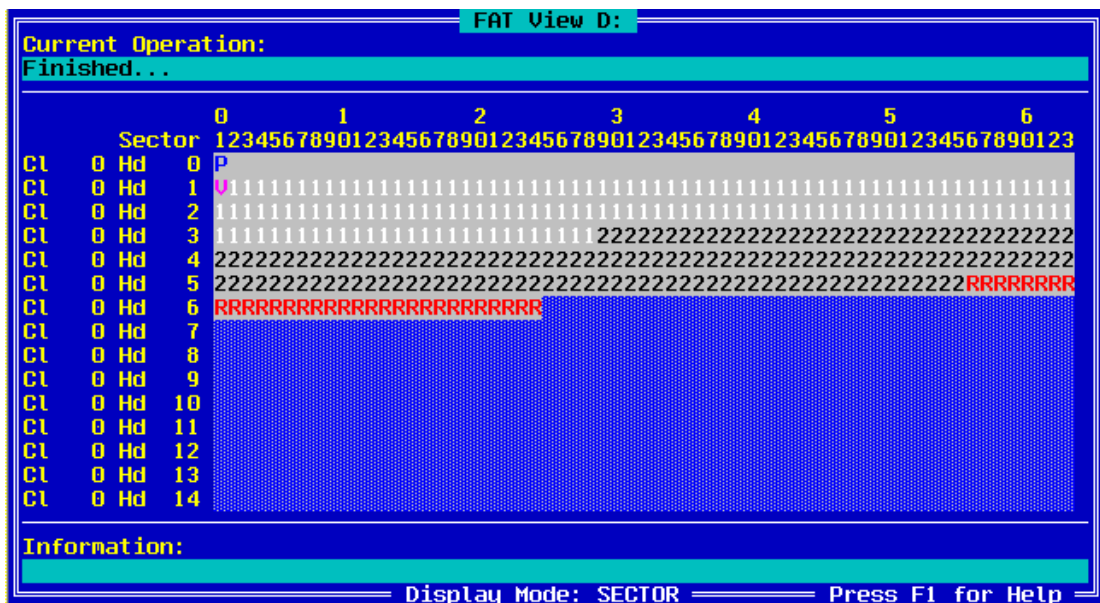
Description

FAT-View displays the entire contents of your FAT partition on a sector by sector basis. All of the information provided by FAT-Info as well as all information related to all files and directories is displayed. The layout of all files and directories, including extended attributes, is presented to you.

You may use the mouse to click on a sector. In the Information window at the bottom of the main window you will be informed of what that particular sector represents. If the sector is part of a file or directory all portions of that file or directory (including attached extended attributes) will be highlighted for you. The directory or file and the number of extents in which the file is in will be displayed for you in the Information window.

As the disk is being scanned, the current file will be displayed in the Current Operation window located at the top of the main window. If the defragment files option has been selected and after the drive has been fully scanned, the files which have qualified for defragmentation will be displayed in the Current Operation window as they are being defragmented. The results of the defragmentation will be displayed in the Information window. The operation of the mouse is automatically disabled as the defragmentation proceeds. Once the defragmentation process has finished, normal control of the mouse is returned.

FAT-View is a multi-threaded application. This means that it allows you to use the arrow keys and the mouse at the same time as the disk is being scanned. You will see a screen similar to this:



for Sector mode displays.



for Cluster mode displays.



for Compressed Cluster mode displays.

If you tell FAT-View to display a non FAT drive you will get the following message:

Drive is not FAT.

-c Clear the free clusters after defragment

The -c switch causes FAT-View to clear all unused clusters after defragmentation of files has been successfully completed.

Note:

This switch is only functional with the -d switch.

-d Defragment files

This switch enables FAT-View to defragment files after it has fully scanned the drive. The current defragmentation option is to move all of the directories to the beginning of the disk, followed by all of the files. The directories and files are not sorted (use DS to do this before starting FAT-View), they are simply laid down in the order in which they were found. Files and directories with the System and Hidden attributes are not moved. FA may be used beforehand to change this. There must be at least one free cluster on a disk for it to be defragmented. To defragment files, FAT-View must also be able to lock the disk for exclusive access. If it can not, FAT-View will inform you of this and then exit.

-g Government security rules for wiping

This switch forces FAT-View to erase the file to the US Department of Defence security specification DoD 5220.22-M. It writes over all of the data area of the file with 1's (FFh) then writes over it all again with 0's (00h). By default, it repeats this procedure 3 times. The optional <n> parameter for this test overrides the default repeat count of 3. Finally, the data area of the file is overwritten with the value of F6h.

Note:

This switch is only functional with the -d switch.

-n Number of buffers; default of 100

The default number of buffers used by FAT-View is 100. The -n switch can be used to change this. However, please note that this value is compared to the

number of free clusters available on the disk. The actual number of buffers is set to the smaller of the two values.

Note:

This switch is only functional with the -d switch.

-r Repeat wipe count

The default wipe count value is 1. This switch specifies a new repeat count. The repeat count is the number of times which the disk is overwritten.

Note:

This switch is only functional with the -c and -d switches.

-u Unattended mode

This switch causes FAT-View to automatically exit when the program has finished. Normally it waits for an ESC or F10 key press to exit.

-v Wipe Value

The default value of the data which is written to the disk is zero (0). Use this switch to specify another value.

Note:

This switch is only functional with the -d switch.

-on Run on <Machine Name>

This switch forces DEPIPE to be loaded and allows FAT-View to access disk drives located on remote machines. The <Machine Name> must be specified with a leading "\\\". The machine name of the remote machine will have been specified when the network software on the remote machine was installed.

Note:

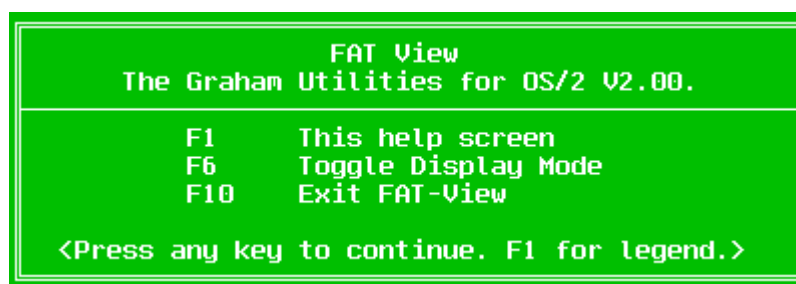
DESVR must be running on the remote machine for this option to work.

Consult Appendix H - DESVR for further details.

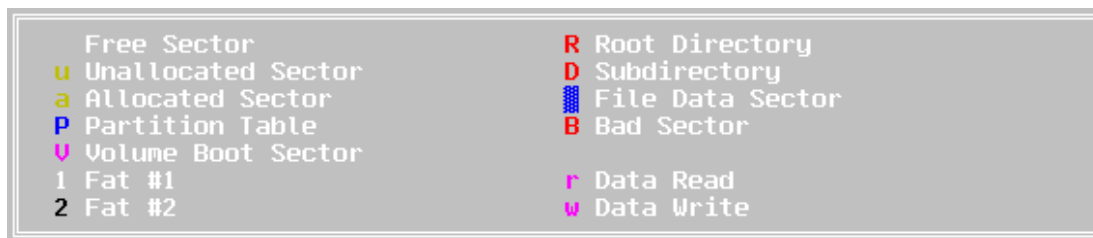
Whilst FAT-View is running, the function keys listed below have the following functions associated with them.

F1 - Help

The help for FAT-View is accessed by pressing the F1 key. You will see this dialog box:



Further details can be further obtained by pressing F1 again:



F6 - Toggle Display Mode

The initial display mode of FAT-View is SECTOR. In this mode the Cylinders, Heads and Sectors are displayed. One sector is displayed per character cell. The F6 key switches between the SECTOR, CLUSTER and COMPRESSED CLUSTER display modes. In the CLUSTER display mode the cluster numbers are displayed. One cluster is displayed per character cell. The COMPRESSED CLUSTER display mode compresses the view of the entire disk into one screen. Each character cell represents <N> clusters. COMPRESSED CLUSTER mode disables the mouse. The mouse can be used to select any character cell in the SECTOR or CLUSTER display modes.

F10 - Exit

The F10 and escape keys exit FAT-View.

Examples

```
FAT-View
```

FAT-View will display the layout of all of the files on the current drive.

```
FAT-View -d E:
```

FAT-View will display the contents of drive E: and then defragment all of the files and directories on that drive.

```
FAT-View E: -du
```

FAT-View will display the contents of drive E:. It will then attempt to defragment all files and directories on that disk and automatically exit when finished.

Note:

Due to the amount of information being display and the frequency at which it is being redisplayed, FAT-View is not recommended for use in OS/2 windowed command sessions. You can speed up the execution of FAT-View by displaying the help as the disk is being scanned.

```
FAT-View C: -on \\WARP
```

FAT-View will display the layout of all files and other information on C: on the remote machine named WARP.

```
FAT-View d: -n100 -dcu -v250 -g
```

FAT-View will display all of the details of drive D:, and then defragment it. Once the defragmentation has finished all of the free clusters will be wiped according to the government wiping rules. The final value left in the clusters will be 250. Once this has completed the program will automatically exit.

FromUNIX - UNIX to OS/2 text conversion utility

Summary

FromUNIX converts UNIX text files to OS/2 text files.

Icon



Command Line Format

Usage: FromUNIX <File In> [File Out]

Description

FromUNIX converts text files from UNIX systems, which use a single line feed character to signal the end of a line, to OS/2 (and DOS) text files which use a carriage return and line feed character pair to indicate the end of line.

FromUNIX takes two parameters. The first <File In> is the file to convert. The second parameter [File Out] is optional. If the second parameter is given, the converted file will be saved to the specified file. If the second parameter is not given, the output is sent to the console. When the output is sent to the console, it may be piped into another program.

ToUNIX is the complement to this program.

Examples

```
FromUNIX ZOO.MAN ZOO.DOC
```

FromUNIX will read the UNIX file ZOO.MAN and convert it to the OS/2 file ZOO.DOC.

```
FromUNIX E:\UNIX\SOURCE\SYSTEM.A E:\OS2\SOURCE\SYSTEM.ASM
```

Convert the file E:\UNIX\SOURCE\SYSTEM.A to the OS/2 file E:\OS2\SOURCE\SYSTEM.ASM.

FromUNIX RFC1035.TXT

FromUNIX will convert the UNIX file RFC1035.TXT and all output will be sent to the console.

FromUNIX RFC1035.TXT | Clipbrd

The UNIX file RFC1035.TXT will be converted and the output will be piped into the system clipboard using the Clipbrd program.

GREP - Text search utility

Summary

GREP is a simple text search utility. It searches for the specified pattern of text in the files specified.

Icon



Command Line Format

Usage: GREP {Switches} <Pattern> <File Spec> <Drive Spec(s)>

Switches

- a All disks.
- c Show the matched characters in Colour.
- f List the File name.
- h Hard disks only.
- i Make the pattern case Insensitive.
- n List the line Number.
- o Only list the file name.
- s Subdirectories.
- x List the lines which do NOT have the pattern in them.

Description

GREP is a simple text search utility. It searches for the specified pattern of text in the files specified. GREP can also be used as a filter - which means that the output from another file or program can be redirected or piped into GREP for it to scan.

This implementation of GREP is more of a FGREP, in that it only does simple text sub string searches - not full regular expression parsing. A full General Regular Expression Parser may be available in a future release.

-a All disks

GREP will scan all logical disk drives defined by the system, starting at drive A:. The drives are scanned in order, starting at A:, then B: and so on.

-c Show the matched characters in Colour

By default, GREP simply displays the line of text on which the pattern match occurred. This switch forces GREP to display the search pattern in green text. It then resets the colour to normal, low intensity white.

-f List the File name

Normally GREP does not display the file name of the file in which the match occurred. This switch displays the file name. It is particularly useful when GREP is used with multiple files (wild cards).

-h Hard disks only

GREP will scan all logical disk drives defined by the system, starting at drive C:. The drives are scanned in order, starting at C:, then D: and so on.

-i Make the pattern case Insensitive

By default, GREP's pattern searches are case sensitive. This switch makes the pattern matches case insensitive. Therefore, with this switch GREP will match the pattern "The" in lines containing "the", "THE", "The" and so on.

-n List the line Number

GREP will display the line number (beginning at one) on which the pattern match occurred.

-o Only list the file name

This option forces GREP to only display the file name of any file which matches the given pattern. This option has precedence over the other options.

-s Subdirectories

GREP will include subdirectories as part of its search of the specified disk drives. If a subdirectory name is included as part of the file specification, then the search will include that subdirectory and all subdirectories below it. If no subdirectory is given as part of the file specification, then the current directory and all below it are included in the search.

-x List the lines that do Not have pattern in them

This switch will force GREP to display every line which does NOT have the pattern in them.

Examples

```
GREP -finc REM C:\CONFIG.SYS
```

If a match is found GREP will display the file name, the line number and any line which has "REM" in it. The pattern search is case insensitive, and the pattern "REM" will be displayed in colour.

```
GREP -fincs LOGON C:\*.CMD
```

GREP will search all .CMD files on drive C: only. It will display the file name, line number, and the case insensitive pattern in colour of any lines which have LOGON in them.

```
GREP -x Test C:\TEST.DAT
```

This will force GREP to display all lines in TEST.DAT which do NOT have the pattern "Test" in them.

```
DIR | GREP -i .EXE
```

This is an example of how GREP can be used as a filter. GREP takes the output of the DIR command and will only display those lines with .EXE in them.

```
DIR | GREP -i .EXE | GREP 12-03-94
```

GREP may be used as a filter multiple times. This example is the same as the above example, but the second invocation of GREP will modify the output so that only .EXE files dated 12-03-94 will be displayed.

```
DIR | GREP -i .EXE | GREP -x 12-03-94
```

This example is similar to the above, except that it will list all .EXE files which are not dated 12-03-94.

```
GREP -fincs SYS H:\FILES.BBS | GREP -ic INFO
```

This will force GREP to list all lines with both SYS and INFO in them. All FILES.BBS files on drive H: will be searched.

This is an example of how GREP can search for multiple keywords within the one line. GREP will report on any line with SYS and INFO in them. The order is not important. It will find matches in and display the following lines:

```
OPSYS_.LZH ( )OPSYS: DOS / OS/2 Utility that informs
SYSLEVEL.LZH ( )Contains format info for the syslevel file
SYSINFO.LZH ( )*** Needs to be described
CNFGSORT.LZH ( )Information file to edit config.sys
CONFIG.LZH ( )NT OS/2 subsystem info on config.sys
OS2APP.LZH ( )info on OS/2 subsystem under NT
INFOBAR1.LZH ( 1)System resource information bar
```

Hexdump - Dump a file in Hexadecimal format

Summary

Hexdump performs a hexadecimal dump of the specified file.

Icon



Command Line Format

Usage: Hexdump {Switches} <File Spec> <Drive Spec(s)>

Switches

-a	All disks.
-c	Clip the high bit.
-h	Hard disks only.
-s	Include subdirectories in the search.
-t	Open the file in TEXT mode. The default is BINARY.
-x	Xor the high bit.

Description

Hexdump will display the contents of files in hexadecimal. The file address is displayed in hexadecimal, as is the file data (in rows of 16 bytes in hexadecimal). An ASCII character representation of the bytes is also displayed on the right of the screen. Any unprintable characters are displayed as a period (".").

If both the -c and -x switches are specified together, the byte is first clipped (-c) and then XORed (-x). The effect of specifying both the -c and -x switches is the same as forcing the high bit on.

-a **All disks**

Hexdump will scan all logical disk drives defined by the system, starting at drive A:. The drives are scanned in order, starting at A:, then B: and so on.

-c Clip the high bit off

This switch strips off the high bit (bit 7) of all bytes which are read. A byte has the range of 0 to 255. This option limits the range to 0 to 127.

-h Hard disks only

Hexdump will scan all logical disk drives defined by the system, starting at drive C:. The drives are scanned in order, starting at C:, then D: and so on.

-s Subdirectories

Hexdump will include subdirectories as part of its search of the specified disk drives. If a subdirectory name is included as part of the file specification, then the search will include that subdirectory and all subdirectories below it. If no subdirectory is given as part of the file specification, then the current directory and all below it are included in the search.

-t Open the file in TEXT mode

The default mode of Hexdump is to open the file in BINARY mode. This means the entire file will be read. In TEXT or ASCII mode the file is opened as a text file. This means that the file will be read until the physical end of file, or an end of file character (26 decimal, Control-Z or 1A hex) is found.

-x Xor the high bit

This switch XORs the high bit (bit 7). All bytes from 0-127 (0-7F hex) will be changed to 128-255 (80-FF hex); and all bytes from 128-255 (80-FF hex) will be changed to 0-127 (0-7F hex).

Examples

```
HEXDUMP *.bin
```

Hexdump will dump any .bin files which it finds in the current directory.

```
HEXDUMP *.ico /s
```

Hexdump will dump any .ico files which it finds in the current directory - as well as any subdirectories below it.

```
HEXDUMP -c TEST.DOC
```

Hexdump the Word Star document TEST.DOC. Word Star documents have the high bit set to indicate the beginning of a word, so the -c switch has been specified.

```
HEXDUMP -x prime.dat
```

Hexdump will display the contents of the file 'prime.dat', after it has XORed bit 7. Some minicomputers (such as the Prime 50 Series) use a character set which is mapped in the upper 128 byte half of the complete 256 byte character set. The -x switch is useful for such files. It has the effect of making the ASCII portion of the dump more readable.

HexEdit - Edit a file in Hexadecimal format

Summary

HexEdit enables you to perform a low level edit of a file - either in hexadecimal or ASCII formats.

Icon



Command Line Format

Usage: HexEdit <File>

Description

HexEdit allows you to edit a file in hexadecimal or in normal ASCII. The data is displayed in both hexadecimal and ASCII. The TAB key switches between hex and ASCII editing modes. When you are editing in hexadecimal, you may only enter hex values ('0'-'9','A'-'F'). In the ASCII editing mode it is possible to use all ASCII keys. The cursor control keys are active in both modes. The Control-Home keys take you to the beginning of the file and the Control-End keys take you to the end. The Page-Up and Page-Down keys move you one page up or down respectively.

The bytes which have been changed are displayed in yellow. You are always prompted before any data is written back to the file. You have the option of discarding the data, saving the data or continuing to edit the data.

F1 - Help

The help for HexEdit is accessed by pressing the F1 key. You will see this dialog box:

```
Hexedit
Edit files in a hexadecimal format.

While editing a file, use the following keys:

^Home      Move to the top of the file.
^End       Move to the bottom of the file.
Page Up    Move up one page.
Page Down  Move down on page.
Enter      Prompt for action.
Tab        Switch between Hex and ASCII edit modes.
Letters    0-9, A-F in Hex mode. All in ASCII mode.
↑,↓,+ or → Move in the appropriate direction.
F2         Toggle Mask state.
F3         Goto address.
F4         Search for data.
F5         Search for next occurrence of data.
Esc, F10   Quit Hexedit.

<Press any key to continue>
```

F2 - Toggle mask state

The F2 key toggles the mask state. When masking is on, only normal ASCII characters are displayed. The default is no mask. With masking off, all characters with the exception of character zero are displayed.

F3 - Go to a location

Use the F3 key to move to a specific location within the file. The following dialog box will be presented to you:

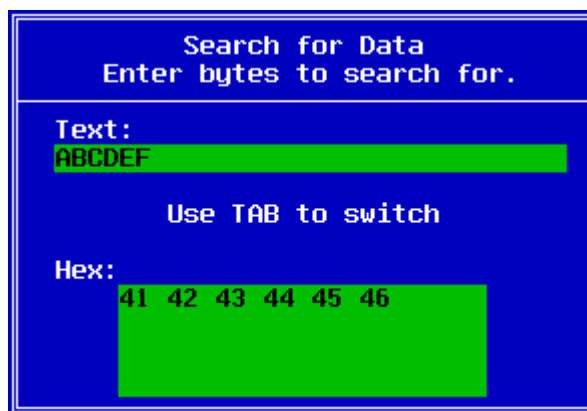
```
Goto location
Enter an address to go to.

Address : 0x0
```

Here you may enter a direct address in hexadecimal. This is particularly useful if you want to move to the middle of a very large file.

F4 - Search for data

The F4 key is used to search for a particular sequence of bytes. You may enter data in ASCII text form or in hexadecimal. The Tab key switches between entering data in ASCII and hexadecimal.



When you have finished entering the bytes, press the Enter key. Pressing Escape aborts the search. If you pressed Enter HexEdit will then begin to search for your data. The search performed is case insensitive. If HexEdit finds a match it will display the address at which the match was found, and then wait for a key to be pressed. It will then take you to that address. If no match was found, HexEdit will tell you that it searched to the end of the data (or file), prompt you for a key press and leave you where you were.

F5 - Continue search

If you wish to search for another occurrence of the same data use the F5 key. If no search data has been entered the F5 key will not continue the search. It will prompt you to enter the data and thus begin the search.

F10 - Exit

The F10 and Escape keys exit the HexEdit program. If needed, you will be prompted to save any unsaved data.

Examples

```
HexEdit charset.dat
```

HexEdit will edit the sample file provided: charset.dat.

```
HexEdit C:\OS2BOOT
```

This will probably fail with the error:

```
SYS0005: Access is denied.
```

This is because, by default, the file OS2BOOT is protected by the Read-Only attribute. HexEdit needs write access to edit a file, and if a file is marked as Read-Only you will get this error. Use FA to change file attributes.

HPFS-Bad - Set bad sectors on HPFS disks

Summary

HPFS-Bad allows you to mark sectors as bad on HPFS partitions.

Icon



Command Line Format

Usage: HPFS-Bad {Switches} Drive Spec <N1> [N2]

Switches

-on Run on <Machine Name>.

Parameters

N1 This is the first sector to be marked as bad. If it is the only parameter given, then only this sector is marked bad. This parameter must be supplied.

N2 This is the last sector to be marked as bad. This is an optional parameter. If it is given, then the first to last sectors (inclusive) will be marked bad.

Note:

Both of the parameters can be given in either decimal or hexadecimal. Decimal is the default. If you wish to enter values in hexadecimal, simply prefix the numbers with a "0x".

Note:

The value of the parameters is in Logical Sectors, not absolute. They start at zero (0) and go up to the maximum as defined by HPFSInfo in the Total number of sectors field.

Description

HPFS-Bad allows you to mark sectors on HPFS disks as bad, so that they will not be used.

The sectors to be marked as bad, must first be free for allocation. If they are already in use HPFS-Bad will report it as such, and the sector will not be marked as bad. HPFSView may be used to determine if the sector is in use, and if so, by what.

-on Run on <Machine Name>

This switch forces DEPIPE to be loaded and allows HPFS-Bad to access disk drives located on remote machines. The <Machine Name> must be specified with a leading "\\\". The machine name of the remote machine will have been specified when the network software on the remote machine was installed.

Note:

DESVR must be running on the remote machine for this option to work.

Consult Appendix H - DESVR for further details.

Examples

HPFS-Bad

This will report an error as no drive specification or starting sector number was given.

HPFS-Bad E: 1000

HPFS-Bad will attempt to mark logical sector 1000 (decimal) as bad.

HPFS-Bad E: 0x1000

HPFS-Bad will attempt to mark logical sector 1000 hexadecimal (4096 decimal) as bad.

HPFS-Bad F: 1000 0x2000

HPFS-Bad will attempt to mark logical sectors 1000 to 8192 (decimal), inclusive as bad.

HPFS-Ext - Report on HPFS file fragmentation

Summary

HPFS-Ext reports on the number of extents (fragments) of files.

Icon



Command Line Format

Usage: HPFS-Ext {Switches} <File Spec> <Drive Spec(s)>

Switches

-e	Extents first
-f	Force file names to lower case.
-h	Hard disks only.
-s	Subdirectories.
-on	Run on <Machine Name>.

Description

HPFS-Ext is a command line utility which reports the number of extents (or fragments) which each file matching the given file specification is in.

HPFS-Ext is capable of reporting on one file, one directory, one drive or all drives in one invocation. This can be especially useful if you wish to check to see if the drive needs defragmentation. The output of HPFS-Ext may be redirected to either a file or a printer to save the output.

If you ask HPFS-Ext to report on a non HPFS drive you will get the following message:

```
Drive is not HPFS.
```

-e Extents first

The default behaviour of HPFS-Ext is to list the file name followed by the number of extents of which each file is comprised. The -e switch forces HPFS-Ext to list the extents first. This allows the output of HPFS-Ext to be piped into smart sort filters (such as GNU's GSORT) which can sort in numeric order. This will give you a list of files in fragmentation order, not file found order.

-f Force file names to lower case

On FAT partitions, file names are always displayed in lower case. HPFS partitions retain their case, but the file system does not differentiate between the case of file names, and they are displayed in the form in which they were entered. The -f option forces all displayed file names to be displayed in lower case.

-h Hard disks only

HPFS-Ext will scan all logical disk drives defined by the system, starting at drive C:. The drives are scanned in order, starting at C:, then D: and so on.

-s Subdirectories

HPFS-Ext will include subdirectories as part of its search of the specified disk drives. If a subdirectory name is included as part of the file specification, then the search will include that subdirectory and all subdirectories below it. If no subdirectory is given as part of the file specification, then the current directory and all below it are included in the search.

-on Run on <Machine Name>

This switch forces DEPIPE to be loaded and allows HPFS-Ext to access disk drives located on remote machines. The <Machine Name> must be specified with a leading "\\\". The machine name of the remote machine will have been specified when the network software on the remote machine was installed.

Note:

DESVR must be running on the remote machine for this option to work.

Consult Appendix H - DESVR for further details.

Examples

HPFS-Ext

HPFS-Ext will report the number of extents of all files in the current directory.

HPFS-Ext /s

HPFS-Ext will report the number of extents of all files in the current directory and all directories below it.

HPFS-Ext -h

HPFS-Ext will scan all hard disks.

HPFS-Ext C:\OS2

HPFS-Ext will list the number of extents of all files in C:\OS2.

HPFS-Ext C:\OS2 /s

HPFS-Ext will list the number of extents of all files in C:\OS2 and all directories below it.

HPFS-Ext E:\Data -on \\MAIN

HPFS-Ext will report the number of extents of all files in E:\DATA on the remote machine named MAIN.

HPFS-UD - Undelete files on HPFS drives

Summary

HPFS-UD allows you to undelete files from HPFS drives, or to recover valid files located on damaged HPFS drives.

Icon



Command Line Format

Usage: HPFS-UD {Switches} <Drive Spec(s)>

Switches

-c	Complete disk scan.
-d	Damaged partition mode.
-f	Keep a log file.
-l	Lock the drive.
-p	Prompt for restore path.
-s	Subdirectories are created.
-z	Full surface scan for root dir FNode.
-on	Run on <Machine Name>.

Description

Note:

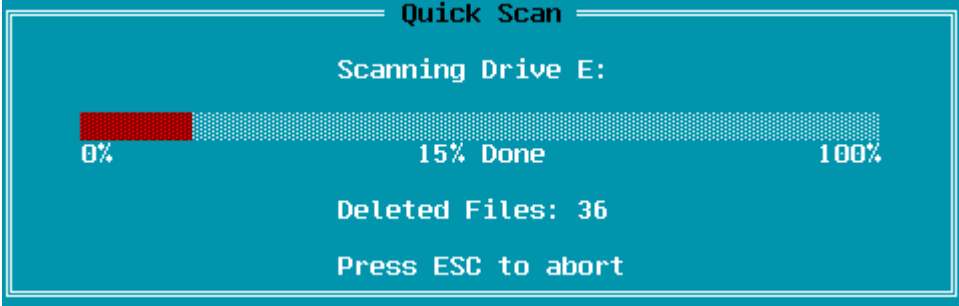
For a complete discussion on the terms and concepts presented here, please consult Appendix G - HPFS Internals.

HPFS-UD scans the disk for deleted FNodes, validates them and then builds up a list to enable you to select the files which you wish to recover.

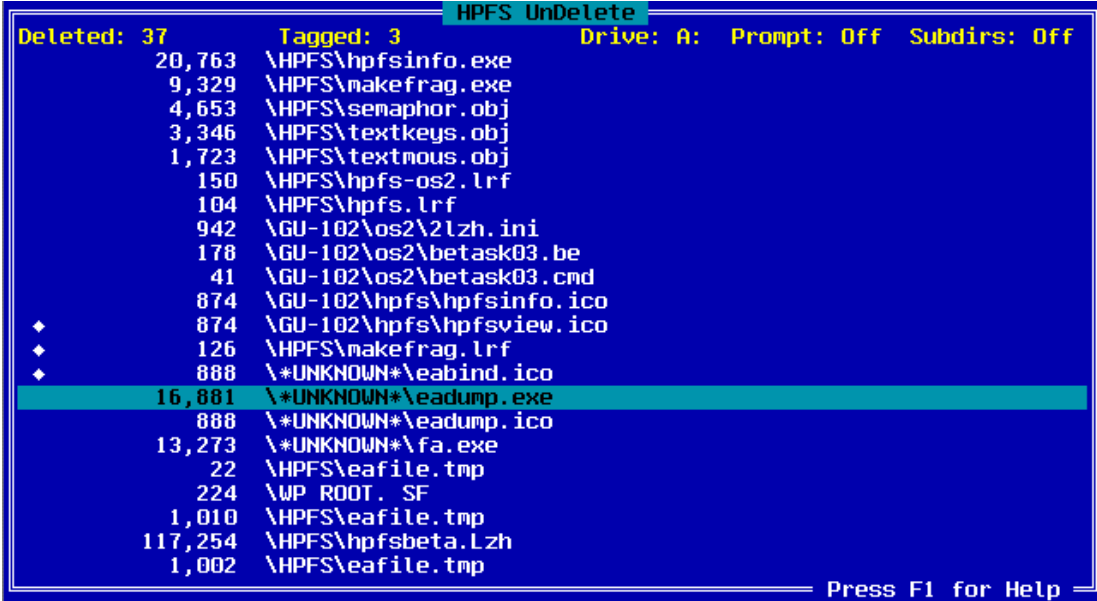
HPFS-UD when undeleting files has two modes of operation: quick scan mode (the default) and complete disk scan. If HPFS-UD is running in quick scan mode when it starts up you will see the following dialog box:

Reading disk information...

It then will begin to scan the disk. As HPFS-UD is scanning the disk, in either quick or complete scan modes, you will see a dialog box similar to:



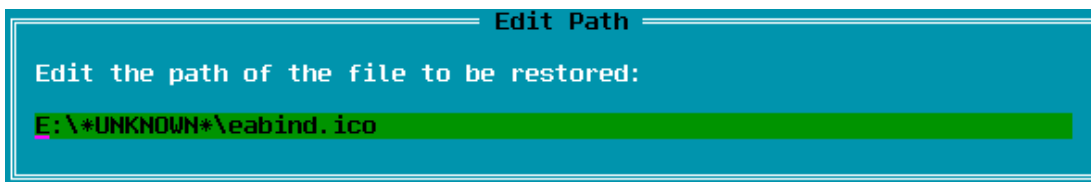
When HPFS-UD has finished scanning the disk you will either see a dialog box informing you that there were no deleted files found, or a window similar to:



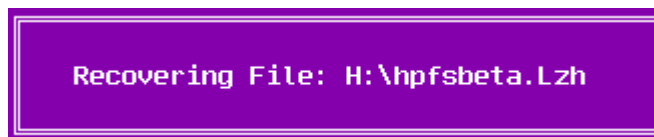
HPFS-UD allows you to select multiple files for recovery in one operation. The Enter key undeletes the currently selected file and Control-Enter allows you to undelete all tagged files. An individual file is tagged using the space bar.

The FNode contains only the first fifteen characters of the filename and the length of the real filename. Any filename that was greater than fifteen characters long will have a '*' character added to the end of the filename.

If you have turned prompting on, or the file to restore already exists, or the filename contains invalid characters, you will be prompted to edit it with a dialog box similar to the following:



As the files are being recovered you will be informed as to which file is currently being recovered.



-c Complete disk scan

The default mode for HPFS-UD is to only scan the deleted and unused file areas - thus making it considerably faster than other HPFS undelete programs. However, there are certain circumstances where you will wish to scan the complete disk. This would happen when the following situation has occurred: Under OS/2 a file can be allocated a length when it is created. Although all of this file may not have been written, it is still marked as owned, and thus not searched by the normal quick scan. There may be deleted files in this space. It is rare that you would wish to use this switch.

-d Damaged partition mode

Under certain circumstances your HPFS partition may become unusable. If you have either run FDISK with unexpected results (you have lost one of your partitions) or it has just become corrupted (through whatever means) and FDISK reports the partition as UNFORMATTED, use this switch. It changes the operation of HPFS-UD to recover valid files from the lost partition.

The first thing which HPFS-UD does is to scan for the root directory FNode. If the root directory FNode is found, then path information can be returned. If it is not, then only the base file name (the first fifteen characters) can be returned. HPFS-UD will then proceed to scan the entire disk and search for good (not deleted) files.

See also the -z switch.

-f Keep a log file

This switch instructs HPFS-UD to maintain a log file of all of its operations and of what ever it finds. It also lists the reasons for which a deleted FNode may have been rejected and thus not added to the list.

The log file is named HPFSUD-<Drive Letter>.LOG. It is created in the directory from which HPFS-UD was started.

-l Lock the drive

This switch attempts to lock the drive so that other processes can not access the drive whilst HPFS-UD is accessing the drive. This is to ensure that no other processes inadvertently overwrite the files which you are trying to recover.

-p Prompt for restore path

By default, the files of the given path are automatically written to the disk. This switch turns on prompting. You will be prompted to edit the complete path and filename of the files as they are being restored.

-s Subdirectories are created

By default, all path information is removed from the filename. This switch turns on subdirectory processing. It leaves the subdirectories intact.

-z Full surface scan for root dir FNode

This switch can only be used in conjunction with the -d switch. By default, the -d switch scans the middle 40% of the disk to find the root directory FNode. If the FNode can not be found in this band, try using this switch. It forces HPFS-UD to scan the complete disk for the root directory FNode.

-on Run on <Machine Name>

This switch forces DEPIPE to be loaded and allows HPFS-UD to access disk drives located on remote machines. The <Machine Name> must be specified with a leading "\\\". The machine name of the remote machine will have been specified when the network software on the remote machine was installed.

Note:

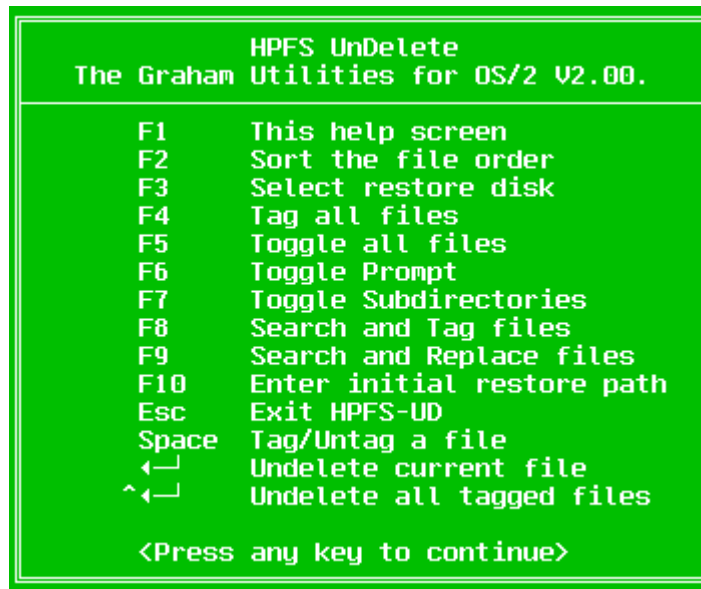
DESVR must be running on the remote machine for this option to work.

Consult Appendix H - DESVR for further details.

Whilst HPFS-UD is running, the function keys listed below have the following functions associated with them.

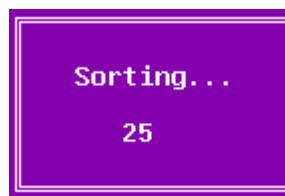
F1 - Help

The help for HPFS-UD is accessed by pressing the F1 key. If the F1 key is pressed you will see the following dialog box:



F2 - Sort the file order

By default, HPFS-UD displays the files in the order in which they are found. The F2 key instructs HPFS-UD to sort the file list in alphabetical order. As the sort progresses you will see this:



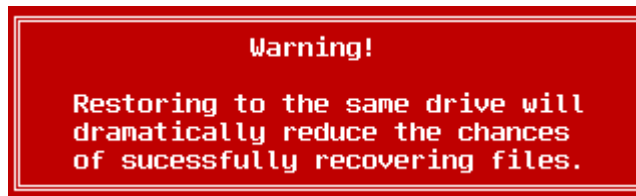
The number which is displayed is the number of entries which have been sorted so far.

F3 - Select restore disk

A: is the default drive to which the deleted files are restored. If you wish to change this, press the F3 key and you will be presented with the select restore drive dialog box:



Use the arrow keys or simply press the required drive letter. If you choose the same drive as that from which you are restoring, you will be warned with the following dialog box:



This is because you are likely to overwrite the file that you are trying to recover!

F4 - Tag all files

The F4 key tags all files which have been found.

F5 - Toggle all files

The F5 key toggles the tagged state of all files.

F6 - Toggle prompt

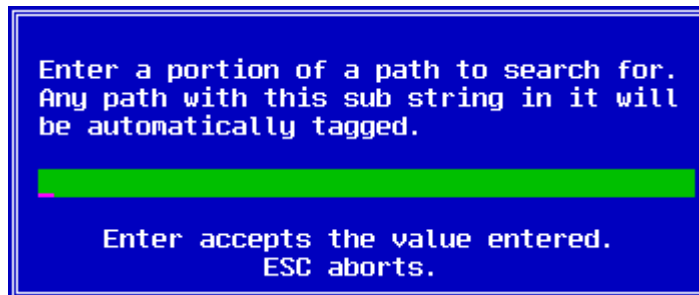
The F6 key toggles the current state of prompting. When prompting is on you will be asked to edit the path and filename of all restored files.

F7 - Toggle subdirectories

The F7 key toggles the current subdirectory state. When subdirectories are on, the path information from the files is not removed. With subdirectories off, all path information is removed and all files are restored into the root directory of the restore drive. All subdirectories are automatically created for you.

F8 - Search and tag files

The F8 key allows you to perform an automatic search and tag function on all of the files which have been found.



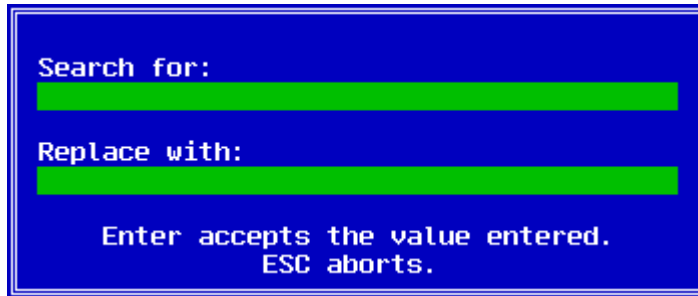
You may enter a sub string which is compared to all of the file names and if there is a match, the file is tagged. This is useful if you wish to recover all files from a particular directory.

F9 - Search and replace files

The F9 key allows you to change the file name and path of the files which have been found. There are two circumstances where this is particularly useful.

First, when the path of the file has been lost and it has been replaced with the *UNKNOWN* path. You may change the *UNKNOWN* path to what it should have been. Changing the *UNKNOWN* path back to a valid one will save considerable time in restoring the files, as you will not have to edit each path individually as it is restored.

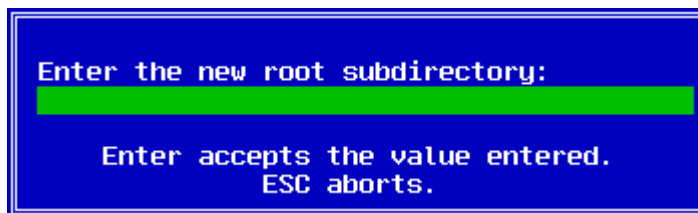
The second circumstance is, if the path itself is incorrect. This may have happened when an old directory FNode was replaced with a newer one. You can use this option to change it back to the correct path.



Enter the characters which you wish to search for and the characters to replace with and press Enter.

F10 - Enter initial restore path

The F10 key allows you to specify an initial restore path off the root directory. This option allows you to graft the restored files to a new path. By default, the files are restored to the paths as they appear. This may cause restored files to be mixed up with normal files. There are circumstances where you may wish to separate the restored files for further examination.



For example, a file "\\MAX\\FILE\\WARPSPEED\\GU-DEMO.LZH" would normally be restored to "\\MAX\\FILE\\WARPSPEED\\GU-DEMO.LZH". If you were to specify a new restore path of "RECOVERED", then using the above example, the restored file would be "\\RECOVERED\\MAX\\FILE\\WARPSPEED\\GU-DEMO.LZH".

Esc - Exit

The escape key exits HPFS-UD.

Examples

HPFS-UD

HPFS-UD will scan for deleted files on the current drive.

HPFS-UD /s

HPFS-UD will scan for deleted files on the current drive and subdirectory processing will initially be turned on.

HPFS-UD H: -d

HPFS-UD will scan drive H: for valid files. This option will work on both valid and damaged volumes.

HPFS-UD D: -l

HPFS-UD will attempt to lock drive D: and search for deleted files. Using this option you will not be able to restore files to the same drive. If HPFS-UD can not lock the drive, it will not continue.

HPFS-UD H: -dz

Scan for valid files on H: and do a full disk scan for the root directory FNode.

HPFS-UD F: -on \\BBS /s

This scans for deleted files on drive F: on the remote machine named BBS. Subdirectory processing will initially be turned on.

HPFSDfrg - Defragment HPFS files

Summary

HPFSDfrg defragments specified files on HPFS volumes.

Icon



Command Line Format

Usage: HPFSDfrg {Switches} <File Spec> <Drive Spec(s)>

Switches

- f Force file names to lower case.
- h Hard disks only.
- o<n> Optimise files with greater than <n> extents.
- r Retry count.
- s Subdirectories.
- on Run on <Machine Name>.

Description

HPFSDfrg is a command line utility which reports the number of extents (or fragments) which each file matching the given file specification is in, and then defragments them if they are in more than a specified number of fragments.

HPFSDfrg is capable of defragmenting one file, one directory, one drive or all drives in one invocation.

If you tell HPFSDfrg to defragment a non HPFS drive you will get the following message:

```
Drive is not HPFS.
```

As HPFSDfrg runs it searches all specified files and determines the number of fragments which each file is in. If it is in a greater number of extents than

specified (a default of two), defragmentation of that file is attempted. For a file to be successfully defragmented there must be enough free space on the drive for a copy of the file. If there is insufficient space on the drive to defragment a particular file you will see the message:

```
No space on drive to defragment <filename>.
```

HPFSDfrg must also be able to obtain exclusive access to the file. If any other processes are accessing a file, HPFSDfrg will not defragment the file.

HPFSDfrg will attempt to defragment a file up to ten times. Sometimes when a file is copied the file actually fragments into more extents than the original file contained. If the file is particularly large or the free space is heavily fragmented (lots of little free spaces as opposed to fewer large ones), the retry count of ten may be exceeded and you will see the message:

```
Free space is too fragmented to allow defragmentation.
```

HPFSDfrg will then continue with the next file.

-d Display defragged files only

This switch causes HPFSDfrg to only display the names of the files that it attempts to defragment. The default operation of HPFSDfrg is to display all filenames and the number of extents that they are in.

-f Force file names to lower case

On FAT partitions, file names are always displayed in lower case. HPFS partitions retain their case, but the file system does not differentiate between the case of file names, and they are displayed in the form in which they were entered. The -f option forces all displayed file names to be displayed in lower case.

-h Hard disks only

HPFSDfrg will scan all logical disk drives defined by the system, starting at drive C:. The drives are scanned in order, starting at C:, then D: and so on.

-o Specify optimisation threshold

By default, HPFSDfrg attempts to defragment files with more than two extents. You may use this switch to change the default. However, if you set the optimisation threshold to too small a number, you may fragment the free spaces. You can use HPFSView and HPFSInfo to check the number of free spaces.

-r Retry Count

The default operation of HPFSDfrg is to attempt to defragment a file up to ten times. If the fragmentation of the file can not be improved upon in ten attempts, HPFSDfrg will issue the free space is too fragmented message. This switch allows you to override the default of ten attempts.

-s Subdirectories

HPFSDfrg will include subdirectories as part of its search of the specified disk drives. If a subdirectory name is included as part of the file specification, then the search will include that subdirectory and all subdirectories below it. If no subdirectory is given as part of the file specification, then the current directory and all below it are included in the search.

-on Run on <Machine Name>

This switch forces DEPIPE to be loaded and allows HPFSDfrg to access disk drives located on remote machines. The <Machine Name> must be specified with a leading "\\\". The machine name of the remote machine will have been specified when the network software on the remote machine was installed.

Note:

DESVR must be running on the remote machine for this option to work.

Consult Appendix H - DESVR for further details.

Examples

HPFSDfrg

HPFSDfrg will attempt to defragment all files in the current directory.

HPFSDFrg /s

HPFSDFrg will attempt to defragment all files in the current directory and all directories below it.

HPFSDFrg -h

HPFSDFrg will attempt to defragment all files with more than two extents on all hard disks.

HPFSDFrg -o3

This changes the optimisation threshold to files with more than three extents and attempts to optimise all files in the current directory.

HPFSDFRG -on \\DATA C:\ /s

HPFSDFrg will attempt to defragment all files with more than two extents on drive C: on the remote machine named DATA.

HPFSInfo - Report on HPFS internals

Summary

HPFSInfo reports all of the information contained in the Super and Spare blocks.

Icon



Command Line Format

Usage: HPFSInfo {Switches} <File Spec> <Drive Spec(s)>

Switches

- b Display bad sector lists.
- c Display the spare directory clusters.
- d Display dynamic hot fix list.
- f Display the free spaces.
- h Hard disks only.
- m Display the bitmap sector lists.
- on Run on <Machine Name>.

Description

Note:

For a complete discussion on the terms and concepts presented here, please consult Appendix G - HPFS Internals.

HPFSInfo displays all of the information which is obtainable from the HPFS Super and Spare blocks.

The information which it returns is:

Super Block Information:

Total number of sectors	This is the total number of useable sectors on the volume. It does not include the hidden sectors which are at the beginning of the volume.
CHKDSK /F was last run	This is the last date and time that CHKDSK /F was run. If CHKDSK has not been run, HPFSInfo will report Never.
Number of bad sectors	This is the total number of bad sectors on the volume.
Root directory FNODE	This is the logical sector number of the root directory FNode.
List of Bitmap sectors	This is the starting sector number of the lists which map out the locations of all of the bitmap sectors.
List of Bad sectors	This is the starting sector of the bad sector lists. It is four sectors long. Each group has room for 511 entries. If there are more than 511 bad sectors, the first entry in the list is a pointer to the next group, and so on.
Number of sectors in directory band	This is the number of sectors in the directory band.
Starting sector of directory band	This is the logical sector number of the first sector of the directory band.
Ending sector of directory band	This is the logical sector number of the last sector of the directory band.
Bitmap sector for directory band	This is the starting logical sector number of the bitmap sectors for the directory band. The directory band has its own bitmap.

ACL information sectors There are eight sectors allocated near the directory band, which are used for holding access control list information. These sectors are only used with HPFS386, but they are reserved and unused in the normal HPFS.

Spare Block Information:

Partition status flag This is the partition status flag. It describes the current status of the disk. The values are added together as follows:

- 00 Partition is Clean.
- 01 Partition is Dirty.
- 02 Spare DIRBLK's are used.
- 04 Hotfix sectors are used.
- 08 Bad sector, corrupt disk.
- 10 Bad bitmap block.
- 20 Partition was quick formatted.
- 40 Reserved.
- 80 An old version wrote to the partition.

List of hot fixes This is the logical sector number of the list which contains all information about the current hot fixes.

Number of hot fixes used This is the number of hot fixes currently in use.

Total number of hot fixes available This is the total number of hotfix locations which have been allocated for the volume.

Number of spare directory clusters This is the total number of spare directory clusters. It is normally 20. They are normally located immediately after the end of the directory band.

Total number of spare dir clusters This is the number of spare directory clusters which are currently free.

Code page directory This is the logical sector number of the code page directory for the volume.

Number of code pages The number of code pages supported on this volume.

Code Page #<n> (Ctry: <x> CP: <y>) at This is the logical sector number of the <n>th code page for the volume. The country code <x> and code page id <y> for the code page is also displayed.

Other Information:

Number of free spaces This is number of free spaces as calculated from the bitmaps.

When the information which is returned is a logical sector number, it is also displayed as a Cylinder, Head and Sector.

If you ask HPFSInfo to report on a non HPFS drive you will get the following message:

Drive is not HPFS.

-b Display bad sector lists

This switch displays the logical sector numbers of all sectors which have been marked as bad. The sector numbers of the bad sector lists are also displayed.

-c Display the spare directory clusters

The -c switch forces HPFSInfo to list the starting logical sector numbers of all of the spare DIRBLKs. Normally there are twenty spare DIRBLKs. Each DIRBLK is four sectors long.

-d Display dynamic hot fix list

This switch tells HPFSInfo to display all of the currently hot fixed sectors. The original bad sector and the sector that it was remapped to are listed.

-f Display the free spaces

The -f switch forces HPFSInfo to display the free space of the disk. The first and last sectors of the free spaces are displayed. They are displayed as both logical sector numbers and as a Cylinder, Head and Sector number.

-h Hard disks only

HPFSInfo will scan all logical disk drives defined by the system, starting at drive C:. The drives are scanned in order, starting at C:, then D: and so on.

-m Display the bitmap sector lists

This switch tells HPFSInfo to display the starting sector locations of all of the bitmap sectors. Each group of bitmap sectors is four sectors long.

-on Run on <Machine Name>

This switch forces DEPIPE to be loaded and allows HPFSInfo to access disk drives located on remote machines. The <Machine Name> must be specified with a leading "\\\". The machine name of the remote machine will have been specified when the network software on the remote machine was installed.

Note:

DESVR must be running on the remote machine for this option to work.

Consult Appendix H - DESVR for further details.

Examples

```
HPFSInfo
```

This will produce the minimum amount of information that HPFSInfo can extract from the Super and Spare blocks.

```
HPFSInfo /b
```

HPFSInfo will display all of its normal information as well as all of the sectors which are currently marked as bad.

```
HPFSInfo -bcdfhm
```

HPFSInfo will scan all hard disks and report on the maximum amount of information available.

```
HPFSInfo C: -bcdfm -on \\WARP
```

HPFSINFO will display all of the information available about C: on the remote machine named WARP.

HPFSNull - Wipe HPFS disks

Summary

HPFSNull wipes HPFS disks on a sector by sector basis.

Icon



Command Line Format

Usage: HPFSNull {Switches} Drive Spec(s)

Switches

- g Government security rules for wiping.
- r Repeat wipe; default of 1.
- v wipe Value; default of 0.
- on Run on <Machine Name>.

Description

HPFSNull wipes all free sectors on a HPFS disk. Sectors which are occupied by files are not wiped. For HPFSNull to be able to process the disk, it must be able to lock the disk and the disk must have a clean status. If the disk is in use by another process, HPFSNull will inform you that it is unable to the lock the disk and exit.

HPFSNull must lock the disk, so that it can be assured that the information which it is reading is accurate. HPFSNull wipes all unallocated data sectors on the disk. It does not wipe any data in the directory band. HPFSNull displays the logical sector number of each sector as it is being wiped. The escape key may be pressed at any point in time to abort the wiping process.

In order to help protect you from accidental erasures you must enter a drive specification.

If HPFSNull is asked to operate on a non HPFS drive you will get the following message:

```
Drive is not HPFS.
```

-g Government security rules for wiping

This switch forces HPFSNull to erase the file to the US Department of Defence security specification DoD 5220.22-M. It writes over all of the data area of the file with 1's (FFh) then writes over it all again with 0's (00h). By default, it repeats this procedure 3 times. The optional <n> parameter for this test overrides the default repeat count of 3. Finally, the data area of the file is overwritten with the value of F6h.

-r Repeat wipe count

The default wipe count value is 1. This switch specifies a new repeat count. The repeat count is the number of times which the disk is overwritten.

-v Wipe Value

The default value of the data which is written to the disk is zero (0). Use this switch to specify another value.

-on Run on <Machine Name>

This switch forces DEPIPE to be loaded and allows HPFSNull to access disk drives located on remote machines. The <Machine Name> must be specified with a leading "\\\". The machine name of the remote machine will have been specified when the network software on the remote machine was installed.

Note:

DESVR must be running on the remote machine for this option to work.

Consult Appendix H - DESVR for further details.

Example

```
HPFSNull F:
```

```
Run HPFSNull on drive F:
```

HPFSStat - Examine the status of HPFS disks

Summary

HPFSStat displays the current status of, and allows modification of the status of HPFS disks.

Icon



Command Line Format

Usage: HPFSStat {Switches} <Drive Spec(s)>

Switches

- c Mark the partition as CLEAN.
- d Mark the partition as DIRTY.
- r Remount the partition.
- on Run on <Machine Name>.

Note:

The -c and -d switches can not be specified together.

If no Drive Spec(s) are entered, the default drive only is searched

Description

HPFSStat allows you to change the status of HPFS disks. If there are no options entered the current status of the HPFS disk is displayed. If the disk was improperly stopped, the status of the disk will most probably be DIRTY. If a disk is DIRTY OS/2 will not mount the drive and you will either have to run CHKDSK /F to clean it, or manually change the status to CLEAN. HPFSStat can be used to manually change the status to CLEAN. HPFSStat can optionally remount the drive, to enable OS/2 to access the files on it. Using the remount option you will not have to reboot the system in order for OS/2 to recognise the drive.

The possible values for the partition status are:

Status Values:

00	Partition is Clean.
01	Partition is Dirty.
02	Spare DIRBLK's are used.
04	Hotfix sectors are used.
08	Bad sector, corrupt disk.
10	Bad bitmap block.
20	Partition was quick formatted.
40	Reserved.
80	An old version wrote to the partition.

If you ask HPFSNull to operate on a non HPFS drive you will get the following message:

```
Drive is not HPFS.
```

-c Mark the partition as CLEAN

This switch forces HPFSStat to mark the partition as CLEAN.

-d Mark the partition as DIRTY

This switch forces HPFSStat to mark the partition as DIRTY.

-r Remount the partition

Normally, after you have CLEANed a partition you will have to reboot the computer in order to allow OS/2 file access to the drive. This option attempts to remount the partition. If a partition is DIRTY OS/2 will not let you access any files on the drive. It will return the error SYS0005: Access is denied. If the disk is successfully remounted OS/2 will allow file access to the drive. Using this option saves you from having to reboot the computer to access the drive.

-on Run on <Machine Name>

This switch forces DEPIPE to be loaded and allows HPFSStat to access disk drives located on remote machines. The <Machine Name> must be specified with a leading "\\\". The machine name of the remote machine will have been specified when the network software on the remote machine was installed.

Note:

DESVR must be running on the remote machine for this option to work.

Consult Appendix H - DESVR for further details.

Examples

```
HPFSStat C: -c
```

HPFSStat will clean the HPFS drive C:.

```
HPFSStat F: -cr
```

HPFSStat will set the status of drive F: to clean and remount the drive.

```
HPFSStat D: -c -on \\DELENN
```

HPFSStat will clean the drive D: on the remote machine named DELENN.

HPFSView - View HPFS drives

Summary

HPFSView allows you to view the contents and layout of your HPFS volumes. As an option it can also visually defragment them.

Icon



Command Line Format

Usage: HPFSView {Switches} <Drive Spec(s)>

Switches

-d	Defragment files.
-o<n>	Optimise files with greater than <n> extents.
-on	Run on <Machine Name>.

Description

Note:

For a complete discussion on the terms and concepts presented here, please consult Appendix G - HPFS Internals.

HPFSView displays the entire contents of your HPFS partition on a sector by sector basis. All of the information provided by HPFSInfo as well as all information related to all files and directories is displayed. The layout of all files and directories, including extended attributes, is presented to you.

You may use the mouse to click on a sector. In the Information window at the bottom of the main window you will be informed of what that particular sector represents. If the sector is part of a file or directory all portions of that file or directory (including attached extended attributes) will be highlighted for you. The directory or file and the number of extents which the file is in will be displayed to you in the Information window.

-d Defragment files

This switch enables HPFSView to defragment files after it has fully scanned the drive.

-o Specify optimisation threshold

The default optimisation threshold of HPFSView is to defragment files with greater than 2 extents. You may use this switch to change this default. However, if you set the optimisation threshold too small, you may fragment the free spaces. You may use HPFSInfo to quickly check the number of free spaces.

-on Run on <Machine Name>

This switch forces DEPIPE to be loaded and allows HPFSView to access disk drives located on remote machines. The <Machine Name> must be specified with a leading "\\\". The machine name of the remote machine will have been specified when the network software on the remote machine was installed.

Note:

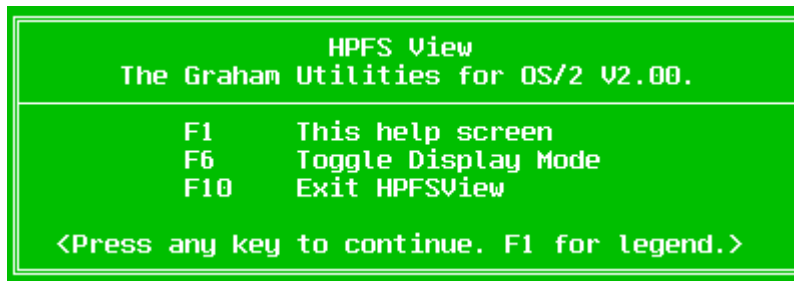
DESVR must be running on the remote machine for this option to work.

Consult Appendix H - DESVR for further details.

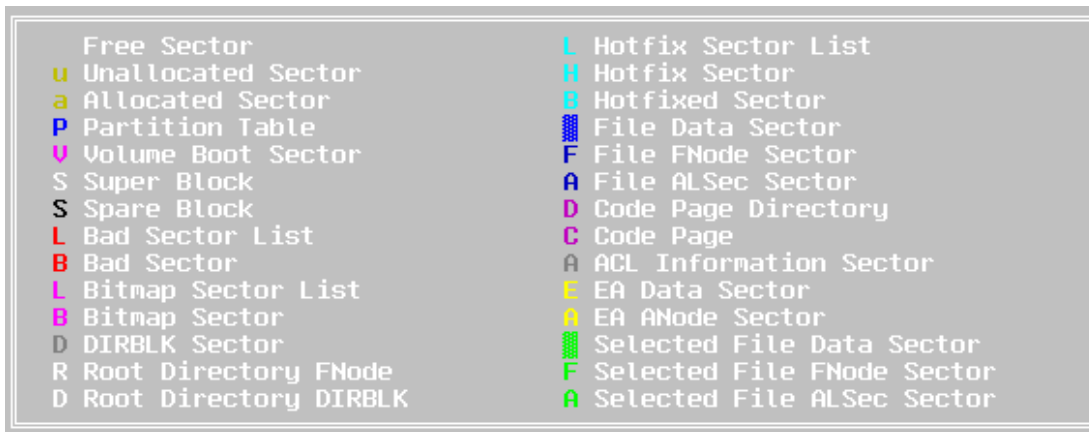
Whilst HPFSView is running, the function keys listed below have the following functions associated with them.

F1 - Help

The help for HPFSView is accessed by pressing the F1 key. You will see this dialog box:



Further details can be obtained by pressing F1 again:



F6 - Toggle Display Mode

The initial display mode of HPFSView is SECTOR. In this mode the Cylinders, Heads and Sectors are displayed. The F6 key toggles between the SECTOR and LSN display modes. In the LSN display mode, the Logical Sector Numbers are displayed.

F10 - Exit

The F10 and escape keys exit HPFSView.

Examples

`HPFSView`

`HPFSView` will display the layout of all of the files on the current drive.

`HPFSView -d E:`

`HPFSView` will display the contents of drive E: and then defragment all of the files which have more than two extents.

`HPFSView E: -d -o1`

`HPFSView` will display the contents of drive E:. It will then attempt to defragment all files with more than one extent.

Note:

Due to the amount of information being displayed and the frequency at which it is being presented, `HPFSView` is not recommended for use in OS/2 windowed command sessions. You can speed up the execution of `HPFSView` by displaying the help dialog box as the disk is being scanned.

`HPFSView C: -on \\WARP`

`HPFSView` will display the layout of all files and other information on C: on the remote machine named WARP.

MIDecode - Decode MIME files

Summary

MIDecode decodes MIME files back into their original binary images.

Icon



Command Line Format

Usage: MIDecode {Switches} <File>

Switches

-e decode Extended Attribute data.

Description

MIDecode decodes MIME files back into their original binary images. MIME format files typically come from the Internet, where their use is increasingly popular - most modern EMail programs support them. MIME files are used to transmit binary files through EMail systems which may only allow 6, 7 or 8 bit data.

MIDecode can decode files which use the UNIX end of line character or normal DOS and OS/2 files. Thus there is no need for a -u switch.

MIME files may have multiple files within them. MIDecode will attempt to decode them all.

The MIME format is similar to the UU and XX encoding, in that it encodes three 8 bit bytes into four 6 bit characters. However, the encapsulation and the character set used is very different.

See MIENCODE for a more complete description of the encoding method.

The SUM utility has been provided to help check the integrity of the decoded files.

Possible Decoding Problems

This is an example of the output of MIEncode:

```
Content-Type: multipart/mixed; boundary="-- part_boundary --"
---- part_boundary ----
Content-Type: text/plain; name="test"
Content-Transfer-Encoding: base64

VGhpcyBpcyBMaW5lICMxDQpUaG1zIG1zIEpibmUgIzINC1RoXMaXMGtGlu
ZSAjMw0KVGhpcyBpcyBMaW5lICM0DQpUaG1zIG1zIEpibmUgIzUNCho
---- part_boundary ----
```

The encoding format is far more relaxed when compared to either the UU or XX encoding formats. The Content-Type line informs us of the type and contents of the file. The first one tells us that there are multiple parts to this file. The most important part of this line is the boundary definition. It is this boundary definition which tells us where one part begins and another starts. When MIDecode produces no output the most common problem is that the boundary definition is missing. If the file came from an EMail package and this information was missing, the View Full Headers option (or similar) may have to be turned on.

-e decode Extended Attribute data

This switch forces MIDecode to look for and decode encoded extended attribute data if any was found. The extended attribute data is automatically bound to the decoded file.

Example

```
MIDECODE GERRY.MIM
```

MIDecode will attempt to decode the file GERRY.MIM.

MIEncode - Encode MIME files

Summary

MIEncode encodes binary files into MIME files.

Icon



Command Line Format

Usage: MIEncode {Switches} <File>

Switches

- c Console output.
- e encode Extended Attribute data.
- s include SUM information in the file.
- t include TABLE information in the file.
- u UNIX compatible file.

Description

MIME Encoding is a scheme which converts 8 bit data such as programs, to a 6 bit format for transmission through 6, 7 or 8 bit (typically electronic mail) networks. Such 6 or 7 bit networks are commonly found in mainframe or UNIX operating system environments.

MIME encoding and MIME decoding of files requires two utilities. MIEncode converts 8 bit data to a 6 bit format. The companion utility, MIDecode, restores the 6 bit data to the original 8 bit image.

A third utility, SUM, may be used to verify that a file has been received and correctly converted.

Encoding Method

MIENCODE breaks a group of three 8 bit characters (24 bits) into four 6 bit characters which are used as an index into the character set table shown below.

The characters are:

```
ABCDEFGHIJKLMNO  
PQRSTUVWXYZabc  
defghijklmnopq  
rstuvwxyz01234  
56789+/  

```

These characters have been especially selected for their ability to remain unchanged when passed through various EMail systems and the character conversions that invariably accompanies them.

Note:

This translation scheme results in a file expansion of approximately 33%.

The extension of the MIME file is .MIM.

MIENCODE also places some information about the original file in the header of the MIME file. The original file name, size, time and date are placed in the header. This is done to provide a little more information to enable the encode/decode process to be checked.

-c Console output

By default, the output of MIEncode is <FILE>.MIM. This switch forces MIEncode to send its output to the console so that it may be redirected or piped into another program.

-e encode Extended Attribute data

The default operation of MIEncode, like every other encoder/decoder pair available, is to ignore the Extended Attribute data associated with files. This switch forces MIEncode to encode any available Extended Attribute data attached to the file and include it in the encoded data stream. The encoding works in the same manner as the actual file data is encoded, but Extended Attribute data is encoded separately at the end of the file and is delimited using an "eaname" keyword as opposed to "name".

-s include SUM information in the file

This switch forces MIENCODE to include SUM information in the top of the MIME file. This is to help the verification process to ensure that the decoded file has been successfully transmitted and decoded. This switch causes MIENCODE to read the file which is to be encoded twice - so it will take slightly longer. The SUM information is identical to the information produced by the SUM utility.

-t include TABLE information in the file

This switch forces MIENCODE to include TABLE information in the top of the MIME file. This is to help the verification process to ensure that the decoded file has been successfully transmitted and decoded. The TABLE information is the actual encoding table which was used to encode the data. If any characters are changed as they travel through various EMail systems, then the same characters in the TABLE should also be changed. Thus the file should still be able to be decoded at the other end. The actual MIME standard does not support this. But if you are having troubles decoding a file, the switch may help determine if the character set has been corrupted or not.

-u UNIX compatible file

Both OS/2 and DOS systems use a carriage return and line feed character pair (CR/LF) to represent an end of line. Under UNIX the standard is to use a single line feed (LF) to represent the end of line. By default MIENCODE produces files which use the CR/LF pair for the end of line. This switch forces MIENCODE to use a single LF character for an end of line.

Examples

```
MIENCODE CHARSET.DAT
```

This will produce a MIME encoded file CHARSET.MIM. It will be a normal OS/2 text file and spaces will be used.

```
MIENCODE CHARSET.DAT /u
```

MIENCODE will produce a UNIX compatible text file called CHARSET.MIM.

```
MIENCODE CHARSET.DAT -stu
```

MIENCODE will produce a UNIX compatible file with quotes. Full checksum information will be included in the MIME file. The encoding table will also be included in the file. Use these options if CHARSET.MIME is likely to go to a UNIX system via an unknown route. The checksum information will help the user validate that the file has been transmitted, received and decoded successfully.

```
MIENCODE CHARSET.DAT -c
```

The encoded data stream will be output to the console.

```
MIENCODE CHARSET.DAT -c | Clipbrd
```

The file CHARSET.DAT will be MIME Encoded and the output will be piped into the Clipbrd program which places the data in the clip board.

```
MIENCODE CHARSET.DAT -ce
```

The file CHARSET.DAT will be MIME Encoded to the console and any Extended Attribute data attached to the file will also be encoded.

NullDisk - Wipe Disk data

Summary

NullDisk protects sensitive data by overwriting either the entire disk or the erased portion of it.

Icon



Command Line Format

Usage: NULLDISK {Switches} Drive Spec(s)

Switches

- a All disks.
- e Erase unused and deleted file areas only.
- g<n> Government security rules for wiping.
- h Hard disks only.
- n No wipe, erase only.
- r<n> Repeat wipe count; default of 1.
- v<n> wipe Value; default of 0.

Description

NULLDISK protects sensitive data by either deleting all files currently on the disk and then overwriting the entire disk, or by overwriting the erased portion of it.

OS/2 when it deletes a file, does not actually remove the data. It only removes the directory entry. NULLDISK physically overwrites the data area of the disk. By default, it deletes all files and then overwrites the data area of the disk. An optional way of running NULLDISK is not to delete all of the files, but simply overwrite the unused portion of the disk. Use this method if you wish to wipe only a few files, or use NULLFILE.

NULLDISK erases the unused portion of the disk by filling it up with a temporary file. Be aware that if other processes are accessing the disk, there may be problems with them as the disk is filled. The temporary file is deleted when the program is finished.

To help protect you from accidental erasures you must enter a drive specification. NULLDISK will now inform you of the intended action and prompt you for confirmation before continuing.

-a All disks

NULLDISK will scan all logical disk drives defined by the system, starting at drive A:. The drives are scanned in order, starting at A:, then B: and so on.

-e Erase unused and deleted file areas only

By default, NULLDISK deletes all files and then overwrites all data areas of the disk. This switch forces NULLDISK to overwrite the unused data area portion of the disk.

-g Government security rules for wiping

This switch forces NULLDISK to erase the disk to the US Department of Defence security specification DoD 5220.22-M. It writes over all of the data area of the disk with 1's (FFh) then writes over it all again with 0's (00h). By default, it repeats this procedure 3 times. The optional <n> parameter for this test overrides the default repeat count of 3. Finally, the data area of the disk is overwritten with the value of F6h.

-h Hard disks only

NULLDISK will scan all logical disk drives defined by the system, starting at drive C:. The drives are scanned in order, starting at C:, then D: and so on.

-n No wipe, erase only

NULLDISK by default, wipes all files and then erases them. This switch will force NULLDISK to only erase (delete) all of the files. No wiping will occur.

-r Repeat wipe count

The default wipe count value is 1. This switch specifies a new repeat count. The repeat count is the number of times that the disk is overwritten.

-v Wipe Value

The default value of the data which is written to the disk is zero (0). Use this switch to specify another value.

Examples

```
NULLDISK
```

This will fail, as a drive specification was not entered.

```
NULLDISK C:
```

This will delete all files and wipe the contents of drive C:.

```
NULLDISK -he
```

NULLDISK will scan all available hard disk drives and wipe the erased file area of each disk.

```
NULLDISK -r4 D:
```

Wipe drive D: 4 times.

```
NULLDISK -v255 -r3 e:
```

Wipe drive E: 3 times, overwriting the disk with the value 255 (FFh) each time.

```
NULLDISK E: -e -g5
```

This will wipe drive G: 5 times using the government security rules - overriding the default of 3. Only the unused portion of the drive will be wiped.

NullFile - Wipe File data

Summary

NULLFILE can either delete files, or protect sensitive data by overwriting it and then deleting it.

Icon



Command Line Format

Usage: NullFile {Switches} File Spec <Drive Specs>

Switches

- a All disks.
- d remove Directories.
- g<n> Government security rules for wiping.
- h Hard disks only.
- n No wipe, erase only.
- p Prompt for wiping or deleting.
- r<n> Repeat wipe count; default of 1.
- s Subdirectories.
- v<n> wipe Value; default of 0.

Description

NULLFILE can either delete files, or overwrite sensitive data and then delete the files.

OS/2 when it deletes a file does not actually remove the data. It only removes the directory entry. NULLFILE physically overwrites the data area of the file. If you have many files to wipe all at the same time, try deleting them and using NULLDISK with the -e switch.

-a All disks

NULLFILE will scan all logical disk drives defined by the system, starting at drive A:. The drives are scanned in order, starting at A:, then B: and so on.

-d Remove directories

NULLFILE will remove the subdirectories which it is told to scan. By default NULLFILE only removes the files in the directories that it encounters. This switch forces NULLFILE to remove the directories as well.

-g Government security rules for wiping

This switch forces NULLFILE to erase the file to the US Department of Defence security specification DoD 5220.22-M. It writes over all of the data area of the file with 1's (FFh) then writes over it all again with 0's (00h). By default, it repeats this procedure 3 times. The optional <n> parameter for this test overrides the default repeat count of 3. Finally, the data area of the file is overwritten with the value of F6h.

-h Hard disks only

NULLFILE will scan all logical disk drives defined by the system, starting at drive C:. The drives are scanned in order, starting at C:, then D: and so on.

-n No wipe, erase only

NULLFILE, by default, wipes the file and then erases it. This switch will force NULLFILE to only erase (delete) the files. No wiping will occur.

-p Prompt for wiping or deleting

By default, NULLFILE will not prompt you before it deletes files. Use this switch if you wish to be prompted before a file is wiped or deleted.

-r Repeat wipe count

The default wipe count value is 1. This switch specifies a new repeat count. The repeat count is the number of times that the file is overwritten.

-s Subdirectories

NULLFILE will include subdirectories as part of its search of the specified disk drives. If a subdirectory name is included as part of the file specification, then the search will include that subdirectory and all subdirectories below it. If no subdirectory is given as part of the file specification, then the current directory and all below it are included in the search.

-v wipe Value

The default value of the data which is written to the disk is zero (0). Use this switch to specify another number.

Examples

```
NULLFILE
```

This is not valid. A file specification must be supplied.

```
NULLFILE *.DOC
```

NullFile will nullify all of the data in all .DOC files in the current directory.

```
NULLFILE *.WKS -H
```

This will instruct NullFile to nullify the contents of all .WKS files on all available hard disk drives.

```
NULLFILE D:\
```

NullFile will wipe the contents of all files in the root directory of Drive D:.

```
NULLFILE D:\ /s
```

NullFile will wipe the contents of all files on drive D:. The directory structures remain intact.

```
NULLFILE D:\ /sn
```

NullFile will delete all files on drive D:. The directory structures remain intact.

```
NULLFILE D:\ /SND
```

NullFile will delete all files on drive D:. The directory structures will be removed.

RestPart - Restore saved partition information

Summary

RestPart allows you to restore information which has been previously saved using SavePart.

Icon



Command Line Format

Usage: RestPart Drive Spec

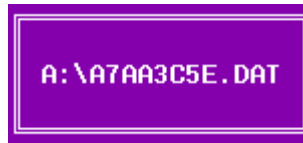
Description

RestPart forms part of the disaster recovery of The Graham Utilities for OS/2. It enables you to restore the partition tables and boot sectors of all local hard disks on your system from the Crash Recovery disk. If the drive was HPFS, then the super and spare blocks may also be restored.

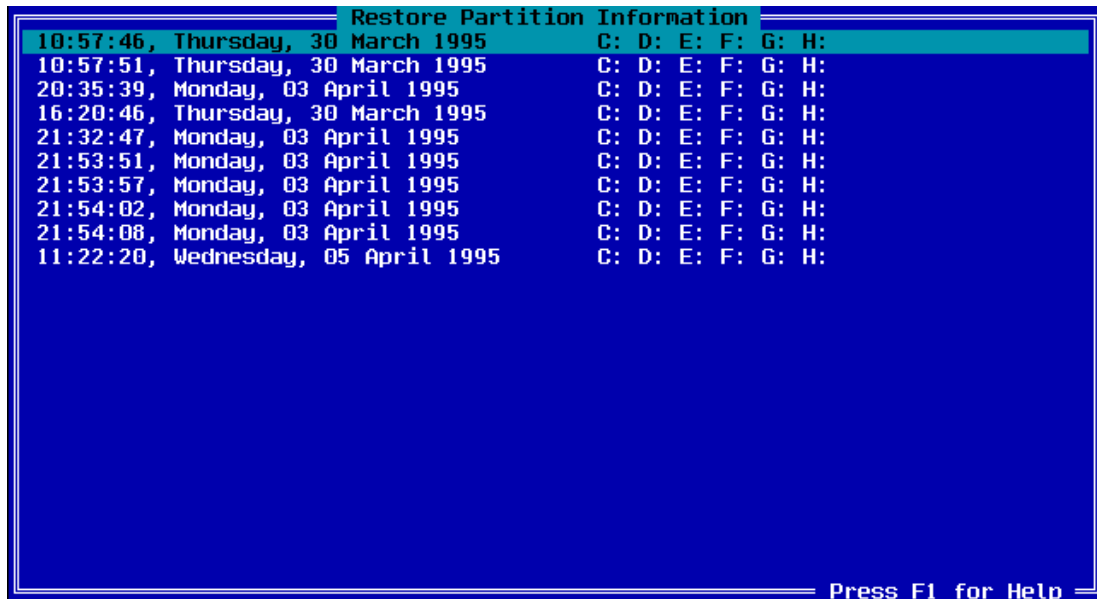
RestPart must be passed the drive letter of the drive which has the Crash Recovery disk in it. If the Crash Recovery disk is not in the specified drive then you will be prompted to insert the correct disk with the following dialog box:



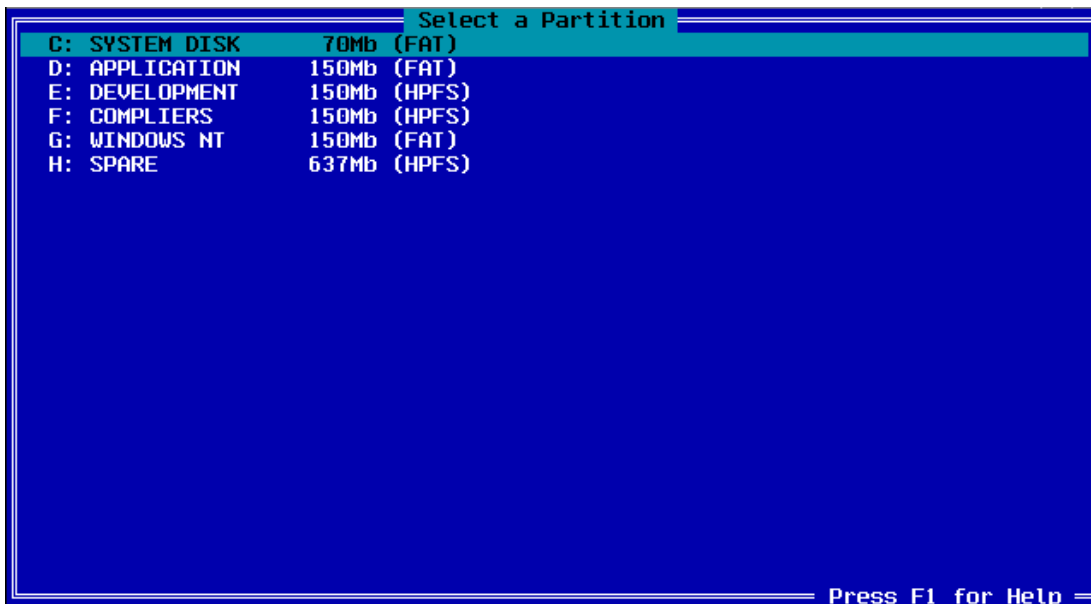
As each of the previously saved information sets is read, the file name of that set is displayed:



When all of the files have been read, you will be presented with a list box similar to the following:



Use the arrow keys to select the file set which you wish to restore. The Enter key selects the highlighted file set. Once you have selected the file set which you wish to restore, you will be presented with a more detailed list of each partition which was saved. It will look similar to:



Each partition is dependent on the previous one. Using the above example, if drive D: was damaged, then D: through to H: would be affected. This means that each partition should be restored, starting at the lowest one. In this instance, drive D: should be restored first. If only D: was damaged, then E: to H: may reappear once D: was correctly restored as E: to H: were not damaged themselves, only the links to them were damaged.

Selecting drive D: you will be presented with this dialog box:

```
Restore Drive D:
Volume Label: APPLICATION
Volume Size : 306Mb
Volume Type : FAT

To restore:

F1 Partition Table

F2 Boot Sector

Esc Aborts restore
F10 Write data to disk
```

This is the dialog box for a FAT partition. If the drive was a HPFS partition then you will see the following dialog box:

```
Restore Drive E:
Volume Label: DEVELOPMENT
Volume Size : 306Mb
Volume Type : HPFS

To restore:

F1 Partition Table

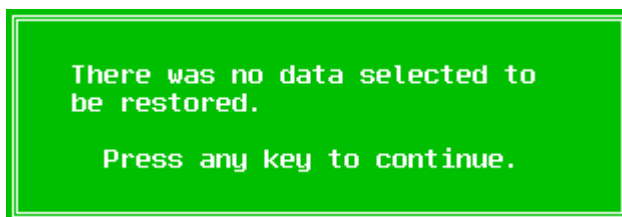
F2 Boot Sector

F3 HPFS Super and Spare blocks

Esc Aborts restore
F10 Write data to disk
```

Using the F1 or F2 keys for FAT partitions or the F1, F2 and F3 keys for HPFS partitions, select the elements which you wish to restore. All available elements are recommended for restoration, however there may be circumstances when you may not wish to restore everything. As each element is selected, an asterisk (*) will appear next to the function key.

Press F10 to write the data to the disk. RestPart will then exit, and the system should then be rebooted in order for the changes to take effect. If no elements were selected for restoration, then you will be prompted by this dialog box:



You will be returned to restore drive dialog box.

Example

RESTPART A:

RestPart reads all of the saved information from the crash recovery disk in drive A:

SavePart - Save partition information

Summary

SavePart saves the partition tables and boot sectors of all of the hard disks in your system. If the drive is a HPFS partition, then the super and spare blocks are also saved.

Icon



Command Line Format

Usage: SavePart {Switches} Drive Spec

Switches

-v Verbose mode

Description

SavePart forms part of the disaster recovery of The Graham Utilities for OS/2. It saves the partition tables and boot sectors of all local hard disks on your system onto the Crash Recovery disk. If the drive is HPFS then the super and spare blocks are also saved.

If you should accidentally FDISK your system and remove some partitions, or an errant program has destroyed your partition tables, your machine will be unusable. If you have saved this information, then RestPart can be used to restore it. SavePart is called as part of the installation process.

It is recommended that you run SavePart BEFORE and AFTER you change your partitions.

SavePart is designed to save all of its information on the Crash Recovery disk which is supplied. The Crash Recovery disk should be in the drive which you specify. If it is not in the drive, then you will be prompted with the following dialog box:



-v Verbose mode

This switch forces SavePart to display the drives which it is saving. By default, it only displays the initial copyright line.

Examples

SavePart A:

This saves the information of all of your hard disks onto the Crash Recovery disk in drive A:

SavePart A: -v

In addition to saving the information of all of your hard disks, SavePart will also display the drives which it is saving. A typical example is:

```
[SavePart, V1.00 - 03/04/95 - (C) Chris Graham - WarpSpeed  
Computers]  
Saving to: A:\401563B6.DAT  
Saving C: SYSTEM DISK      70Mb (FAT)  
Saving D: APPLICATION      150Mb (FAT)  
Saving E: DEVELOPMENT      150Mb (HPFS)  
Saving F: COMPLIERS        150Mb (HPFS)  
Saving G: WINDOWS NT      150Mb (FAT)  
Saving H: SPARE             637Mb (HPFS)
```

Space - Disk Drive Space

Summary

Space will display the total and the free sizes of all disks which it can locate, both local and remote.

Icon



Command Line Format

Usage: Space {Switches}

Switches

-c	Colour warning display
-g	Graphic representation
-p	Pause between graphic screens

Description

Space displays some information about all disk drives. It will display the type, either Local or Remote (a network drive), the file system type, the volume label, the total disk size and the available amount of free disk space. It will also list the total size and the total free disk space of all disks.

This is a typical example:

Type	System	Label	Total	Free	
LOCAL	C:	FAT	SYSTEM DISK	73,222,144	14,714,880
LOCAL	D:	FAT	APPLICATION	157,097,984	14,573,568
LOCAL	E:	HPFS	DEVELOPMENT	157,270,016	118,109,184
LOCAL	F:	HPFS	COMPILERS	157,270,016	30,936,576
LOCAL	G:	FAT	TEST	157,097,984	157,097,984
LOCAL	H:	HPFS	SPARE	667,926,528	216,890,368
LOCAL	I:	CDFS	OS2_CD_ROM	321,073,152	0
REMOTE	J:	LAN	BBS-C	212,097,024	97,678,336
REMOTE	K:	LAN	BBS-D	85,202,944	82,528,768
			Local Total	1,690,957,824	552,322,560
			Remote Total	297,299,968	180,207,104
			Total	1,988,257,792	732,529,664

-c Colour warning display

The default operation of the -g switch is to display a bar graph of the disk usage. The bar is displayed in bright green.

The -c switch modifies this default behaviour to include colour warning information in the colour of the bar graph. The colours are broken down into the following bands:

% Full	Colour of Bar Graph
0 - 75%	Green
76 - 89%	Yellow
90 - 100%	Red

Note:

This switch is only meaningful with the -g switch.

-g Graphic representation

The default mode of Space is to list the total and free sizes in numbers. This switch shows you a bar graph indicating how much of the disk is used.

-p Pause between graphic screens

Because the -g switch may produce many screen fulls of information, this switch prompts you to press a key between each drive as it is listed.

Examples

SPACE

Space will display the sizes in numbers.

SPACE -g

Space will display its information using a bar graph.

SPACE -gp

Space will display its information using a bar graph and pause between each disk drive.

SUM - Check Sum Files

Summary

Sum produces a check sum of a file using a variety of methods.

Icon



Command Line Format

Usage: SUM {Switches} <File Spec> <Drive Spec(s)>

Switches

- a All disks.
- h Hard disks only.
- s Include subdirectories in the search.

Note:

If no file specification is entered, all files are listed. If drive specifications are entered then those drives are searched. If no drive specifications are entered then the current drive is searched.

Description

Sum produces a check sum of all files matching the given file specification using a variety of methods. It will display a check sum using the following methods:

- SUM8 All of the bytes are added up using an 8 bit (or byte) number. This effectively lists the sum of bytes modulo 255 (FFh). The value is displayed in hexadecimal.
- SUM16 All of the bytes are added up using an 16 bit (or word) number. This effectively lists the sum of bytes modulo 65,525 (FFFFh). The value is displayed in hexadecimal.

SUM32	All of the bytes are added up using an 32 bit (or double word) number. This effectively lists the sum of bytes modulo 4,294,967,295 (FFFFFFFFh). The value is displayed in hexadecimal.
CRC16	This method uses a special 16 bit polynomial to calculate the displayed result. The value is displayed in hexadecimal.
CRC32	This method uses a special 32 bit polynomial to calculate the displayed result. The value is displayed in hexadecimal.
BSD	This method is compatible with the SUM command using BSD UNIX. The value is displayed in decimal.
SysV	This method is compatible with the SUM command using System V UNIX. It is the same as SUM32, except that the value is displayed in decimal.

As SUM is generally used by UNIX systems, SUM is normally used with MIDecode, UUDecode or XXDecode to verify the integrity of the file. Quite often, the original encoding program will place the SUM of the original file in the UU file.

-a All disks

SUM will scan all logical disk drives defined by the system, starting at drive A:. The drives are scanned in order, starting at A:, then B: and so on.

-h Hard disks only

SUM will scan all logical disk drives defined by the system, starting at drive C:. The drives are scanned in order, starting at C:, then D: and so on.

-s Subdirectories

SUM will include subdirectories as part of its search of the specified disk drives. If a subdirectory name is included as part of the file specification, then the search will include that subdirectory and all subdirectories below it. If no subdirectory is given as part of the file specification, then the current directory and all below it are included in the search.

Examples

```
SUM *.c
```

SUM will look for all .C files and list the 8, 16 and 32 bit sum, 16 and 32 bit CRC, BSD and System V values for each of the files.

```
SUM C:\CONFIG.SYS
```

SUM the file C:\CONFIG.SYS only.

ToUNIX - OS/2 to UNIX text conversion utility

Summary

ToUNIX converts OS/2 text files to UNIX text files.

Icon



Command Line Format

Usage: ToUNIX <File In> [File Out]

Description

ToUNIX converts text files from OS/2 (and DOS) systems (which use a carriage return and line feed character pair to indicate the end of line) to UNIX systems (which use a single line feed character to signal the end of a line).

ToUNIX takes two parameters. The first <File In> is the file to convert. The second parameter [File Out] is optional. If the second parameter is given, the converted file will be saved to the specified file. If the second parameter is not given, the output is sent to the console. When the output is sent to the console, it may be piped into another program.

FromUNIX is the complement to this program.

Examples

```
ToUNIX ZOO.DOC ZOO.MAN
```

ToUNIX will read the OS/2 file ZOO.DOC and convert it to the UNIX file ZOO.MAN.

```
ToUNIX E:\OS2\SOURCE\SYSTEM.ASM E:\UNIX\SOURCE\SYSTEM.A
```

Convert the file E:\OS2\SOURCE\SYSTEM.ASM to the UNIX file E:\UNIX\SOURCE\SYSTEM.A.


```
ToUNIX RFC1035.TXT
```

ToUNIX will convert the OS/2 file RFC1035.TXT and all output will be sent the console.

```
ToUNIX RFC1035.TXT | Clipbrd
```

The OS/2 file RFC1035.TXT will be converted and the output will be piped into the system clipboard using the Clipbrd program.

UUDecode - Decode UU files

Summary

UUDecode decodes UU files back into their original binary image.

Icon



Command Line Format

Usage: UUDecode {Switches} <File>

Switches

- e decode Extended Attribute data.
- q do not use Quote character (") instead of space.

Description

UUDecode decodes UU files back into their original binary images. UU files typically come from UNIX systems. They are used to transmit binary files through EMail systems which may only allow 6, 7 or 8 bit data.

UUDecode can decode files which use the UNIX end of line character or normal DOS and OS/2 files. Thus, there is no need for a -u switch.

UU files may have multiple files within them. UUDecode will attempt to produce them all.

UUDecode will ignore any lines in the UU file until reaching the "begin" line. This means UUDecode is useful in reading UU files which have EMail headers contained within them.

See UUENCODE for a more complete description of the encoding method.

The SUM utility has been provided to help check the integrity of the decoded files.

-e decode Extended Attribute data

This switch forces UUDecode to look for and decode any encoded extended attribute data it has found. The extended attribute data is automatically bound to the decoded file.

-q do not use Quote character (') instead of space

This is the current form of UU encoding. It uses a single quote character (') to replace all space characters in the encoded data stream. This is to help avoid problems with space compression. This switch turns off the space replacement with quotes.

Examples

```
UUDECODE 1542B.ROM
```

UUDecode will attempt to decode the file 1542B.ROM.

```
UUDECODE -q OLDFILE.UUE
```

UUDecode will attempt to decode the file OLDFILE.UUE using the older encoding method.

UUEncode - Encode UU files

Summary

UUEncode encodes binary files into UU files.

Icon



Command Line Format

Usage: UUEncode {Switches} <File>

Switches

- c Console output.
- e encode Extended Attribute data.
- q do not use Quote character (") instead of space.
- s include SUM information in the file.
- t include TABLE information in the file.
- u UNIX compatible file.

Description

UUEncoding is a scheme which converts 8 bit data such as programs, to a 6 bit format for transmission through 6, 7 or 8 bit (typically electronic mail) networks. Such 6 or 7 bit networks are commonly found in mainframe or UNIX operating system environments.

The UUEncoding and UUDecoding of files requires two utilities. These two utilities are UUDecode and UUEncode. UUEncode converts 8 bit data to a 6 bit format. The companion utility, UUDecode, restores the 6 bit data to the original 8 bit image.

A third utility, SUM, may be used to verify that a file has been received and correctly converted.

Encoding Method

UUENCODE breaks a group of 3 eight bit characters (24 bits) into 4 six bit characters and then add 32 (a space) to each six bit character which maps it into the readily transmittable character set.

The characters are:

```
! "#$%&'()*+,-./
0123456789:;<=>?
@ABCDEFGHIJKLMNO
PQRSTUVWXYZ[\]^_
```

Note:

This translation scheme results in a file expansion of approximately 33%.

The extension of the UU file is .UUE.

UUENCODE also places some information about the original file in the header of the UU file. The original file name, size, time and date are placed in the header. This is done to provide more information to enable the encode/decode process to be checked.

-c Console output

By default, the output of UUEncode is <FILE>.UUE. This switch forces UUEncode to send its output to the console so that it may be redirected or piped into another program.

-e encode Extended Attribute data

The default operation of UUEncode, like every other encoder/decoder pair available, is to ignore the Extended Attribute data associated with files. This switch forces UUEncode to encode any available Extended Attribute data attached to the file and include it in the encoded data stream. The encoding works in the same manner as the actual file data which is encoded - but Extended Attribute data is encoded separately at the end of the file and is delimited using the "eabegin" keyword as opposed to "begin".

-q do not use Quote character (") instead of space

There is a more recent encoding method which uses a single quote character (') instead of a space. This switch forces UUENCODE to replace all spaces in the encoded character stream with quote characters. This is because some transmission mechanisms compress or remove spaces - so the spaces are replaced with quotes.

-s include SUM information in the file

This switch forces UUENCODE to include SUM information in the top of the UU file. This is to help the verification process to ensure that the decoded file has been successfully transmitted and decoded. This switch causes UUENCODE to read the file which is to be encoded twice - so it will take slightly longer. The SUM information is identical to the information produced by the SUM utility.

-t include TABLE information in the file

This switch forces UUENCODE to include TABLE information in the top of the UU file. This is to help the verification process to ensure that the decoded file has been successfully transmitted and decoded. The TABLE information is the actual encoding table which was used to encode the data. If any characters are changed as they travel through various EMail systems, then the same characters in the TABLE should also be changed. So the file should still be able to be decoded at the other end.

-u UNIX compatible file

Both OS/2 and DOS systems use a carriage return and line feed (CR/LF) character pair (CR/LF) to represent an end of line. Under UNIX the standard is to use a single line feed (LF) to represent the end of line. By default UUENCODE produces files which use the CR/LF pair for the end of line. This switch forces UUENCODE to use a single LF character for an end of line.

Examples

```
UUENCODE CHARSET.DAT
```

This will produce a UU encoded file called CHARSET.UUE. It will be a normal OS/2 text file and spaces will be used.

```
UUENCODE CHARSET.DAT /u
```

UUENCODE will produce a UNIX compatible text file called CHARSET.UUE. Quote characters will be used.

```
UUENCODE CHARSET.DAT /q
```

This will produce CHARSET.UUE - a normal OS/2 text file with spaces instead of quotes.

```
UUENCODE CHARSET.DAT /u
```

UUENCODE will produce a UNIX compatible file with quotes. Use these options if CHARSET.UUE is likely to go to a UNIX system.

```
UUENCODE CHARSET.DAT -stu
```

UUENCODE will produce a UNIX compatible file, with quotes. Full checksum information will be included in the UU file. The encoding table will also be included in the file.

Use these options if CHARSET.UUE is likely to go to a UNIX system via an unknown route. The checksum information will help the user validate that the file has been transmitted, received and decoded successfully.

```
UUENCODE CHARSET.DAT -c
```

The encoded data stream will be displayed on the console.

```
UUENCODE CHARSET.DAT -c | Clipbrd
```

The file CHARSET.DAT will be UUEncoded and the output will be piped into the Clipbrd program which places the data in the clip board.

```
UUENCODE CHARSET.DAT -ce
```

The file CHARSET.DAT will be UUEncoded to the console and any Extended Attribute data attached to the file will also be encoded.

XXDecode - Decode XX files

Summary

XXDecode decodes XX files back into their original binary image.

Icon



Command Line Format

Usage: `XXDecode {Switches} <File>`

Switches

`-e` decode Extended Attribute data.
`-q` do not use Quote character (") instead of space.

Description

XXDecode decodes XX files back into their original binary images. XX files typically come from UNIX systems. They are used to transmit binary files through EMail systems which may only allow 6, 7 or 8 bit data.

XXDecode can decode files which use the UNIX end of line character or normal DOS and OS/2 files. Thus there is no need for a `-u` switch.

XX files may have multiple files within them. XXDecode will attempt to produce all of them.

XXDecode will ignore any lines in the XX file until the "begin" line is reached. This makes XXDecode useful when reading XX files which contain EMail headers in them.

See `XXENCODE` for a more complete description of the encoding method.

The `SUM` utility has been provided to help check the integrity of the decoded files.

-e decode Extended Attribute data

This switch forces XXDecode to look for and decode any encoded extended attribute data it may find. The extended attribute data is automatically bound to the decoded file.

-q do not use Quote character (") instead of space

The current form of XX encoding uses a single quote character (') to replace all space characters in the encoded data stream. This is to help avoid problems with space compression. This switch turns the space replacement with quotes off.

Examples

```
XXDECODE 1542B.ROM
```

XXDecode will attempt to decode the file 1542B.ROM.

```
XXDECODE -q OLDFILE.XXE
```

XXDecode will attempt to decode the file OLDFILE.XXE using the older encoding method.

XXEncode - Encode XX files

Summary

XXEncode encodes binary files into XX files.

Icon



Command Line Format

Usage: XXEncode {Switches} <File>

Switches

- c Console output.
- e encode Extended Attribute data.
- q do not use Quote character (") instead of space.
- s include SUM information in the file.
- t include TABLE information in the file.
- u UNIX compatible file.

Description

XXEncoding is a scheme which converts 8 bit data, such as programs, to a 6 bit format for transmission through 6, 7 or 8 bit (typically electronic mail) networks. Such 6 or 7 bit networks are commonly found in mainframe or UNIX operating system environments.

The XXEncoding and XXDecoding of files requires two utilities. These two programs are XXDecode and XXEncode. XXEncode converts 8 bit data to a 6 bit format. The companion utility, XXDecode, restores the 6 bit data to the original 8 bit image.

A third utility, SUM, may be used to verify that a file has been received and correctly converted.

Encoding Method

XXENCODE breaks a group of 3 eight bit characters (24 bits) into 4 six bit characters and then adds 32 (a space) to each six bit character which maps it into the readily transmittable character set. It is basically identical to UU encoding except that the base encoding character set is different. XX encoding came after UU encoding due to problems which UU encoding had with certain character set translation schemes - most notably EBCDIC.

The characters are:

```
+ - 0123456789ABCD
EFGHIJKLMNOPQRST
UVWXYZabcdefghijklmnop
klmnopqrstuvwxyz
```

Note:

This translation scheme results in a file expansion of approximately 33%.

The extension of the XX file is .XXE.

XXENCODE also places some information about the original file in the header of the XX file. The original file name, size, time and date are placed in the header. This is done to provide enough information to enable the encode/decode process to be checked.

-c Console output

By default, the output of XXEncode is <FILE>.XXE. This switch forces XXEncode to send its output to the console to enable it to be redirected or piped into another program.

-e encode Extended Attribute data

The default operation of XXEncode, like every other encoder/decoder pair available, is to ignore the Extended Attribute data associated with files. This switch forces XXEncode to encode any available Extended Attribute data attached to the file and include it in the encoded data stream. The encoding works in the same manner that the actual file data is encoded, but Extended Attribute data is encoded separately at the end of the file and is delimited using the "eabegin" keyword as opposed to "begin".

-q do not use Quote character (") instead of space

There is a more recent encoding method which uses a single quote character (') instead of a space. This switch forces XXENCODE to replace all spaces in the encoded character stream with quote characters. This is because some transmission mechanisms compress or remove spaces. Thus, the spaces are replaced with quotes.

-s include SUM information in the file

This switch forces XXENCODE to include SUM information in the top of the XX file. This is to help the verification process to ensure that the decoded file has been successfully transmitted and decoded. This switch causes XXENCODE to read the file which is to be encoded twice, so it will take slightly longer. The SUM information is identical to the information produced by the SUM utility.

-t include TABLE information in the file

This switch forces XXENCODE to include TABLE information in the top of the XX file. This is to help the verification process to ensure that the decoded file has been successfully transmitted and decoded. The TABLE information is the actual encoding table which was used to encode the data. If any characters are changed as they travel through various EMail systems then the same characters in the TABLE should also be changed. Thus the file should still be able to be decoded at the other end.

-u UNIX compatible file

Both OS/2 and DOS systems use a carriage return and line feed (CR/LF) character pair to represent an end of line. Under UNIX the standard is to use a single line feed (LF) to represent the end of line. By default, XXENCODE produces files which use the CR/LF pair for the end of line. This switch forces XXENCODE to use a single LF character for an end of line.

Examples

```
XXENCODE CHARSET.DAT
```

This will produce a XX encoded file CHARSET.XXE. It will be a normal OS/2 text file and spaces will be used.

```
XXENCODE CHARSET.DAT /u
```

XXENCODE will produce a UNIX compatible text file called CHARSET.XXE. Space characters will be used.

```
XXENCODE CHARSET.DAT /q
```

This will produce CHARSET.XXE - a normal OS/2 text file with spaces instead of quotes.

```
XXENCODE CHARSET.DAT /u
```

XXENCODE will produce a UNIX compatible file with quotes. Use these options if CHARSET.XXE is likely to go to a UNIX system.

```
XXENCODE CHARSET.DAT -stu
```

XXENCODE will produce a UNIX compatible file with quotes. Full checksum information will be included in the XX file. The encoding table will also be included in the file. Use these options if CHARSET.XXE is likely to go to a UNIX system via an unknown route. The checksum information will help the user validate that the file has been transmitted, received and decoded successfully.

```
XXENCODE CHARSET.DAT -c
```

The encoded data stream will be output to the console.

```
XXENCODE CHARSET.DAT -c | Clipbrd
```

The file CHARSET.DAT will be XXEncoded and the output will be piped into the Clipbrd program which places the data in the clip board.

```
XXENCODE CHARSET.DAT -ce
```

The file CHARSET.DAT will be XXEncoded to the console and any Extended Attribute data attached to the file will also be encoded.

Appendix A - Extended Attributes

Extended Attributes - Overview

Extended Attributes are a method which OS/2 uses to store information about a file without modifying the contents of the file. The information is not stored within the file itself. This applies equally to directories.

The file system itself stores this information. On a FAT partition the data is stored in a file called "EA DATA .SF". This is normally a hidden, system file in the root directory. A HPFS partition stores the data in a different manner. The HPFS file system has separate structures within the file system which allow for the storage of extended attribute data to be separate and distinct from the file data. To a program there is no difference in how this data is manipulated, and the data itself is identical.

The DIR command on a FAT partition will report something similar to:

```
The volume label in drive C is SYSTEM DISK.  
The Volume Serial Number is 3A1F:1BEB  
Directory of C:\
```

```
IO          SYS      33430   9-04-91   5:00  
MSDOS       SYS      37394   9-04-91   5:00  
OS2         <DIR>    28-07-93 20:45  
EA DATA    SF       940032  28-07-93 20:47  
WP ROOT     SF        224    3-01-94 20:09  
STARTUP     CMD       118    3-01-94 19:24
```

With the DIR command on any other type of partition (HPFS, LAN, CD-ROM etc) you will notice that you get an extra column of numbers. This extra column is the size of the Extended Attributes associated with that file or directory.

```
The volume label in drive E is DEVELOPMENT.  
The Volume Serial Number is A608:DC15  
Directory of E:\
```

```
21-12-93   20:38      <DIR>          0  .  
21-12-93   20:38      <DIR>          0  ..  
10-01-93   0:48       37754         0  ALLFILES.LST  
23-12-92   21:15     234402        0  OLVR11G.LZH  
 6-01-94   0:00      <DIR>     10858  OS2UTILS
```

It can be seen from the above that the directory OS2UTILS has 10,858 bytes of extended attributes associated with it.

The size of extended attributes can be displayed on FAT partitions by using the DIR /N command.

Extended Attributes - Details

Each extended attribute has two parts - a name and data. The name is just a string and the data can be anything. There is a limit of 64K of extended attribute data per file. There is no limit to the number of extended attributes per file - so long as the 64K restriction is observed.

There are some predefined types of extended attributes, although applications may define and use their own. The predefined attributes are listed below:

EAT_BINARY	The data is binary. The first word is the length.
EAT_ASCII	The data is ASCII text. The first word is the length.
EAT_BITMAP	The data is a BITMAP. The first word is the length.
EAT_METAFILE	The data is a METAFILE. The first word is the length.
EAT_ICON	The data is an ICON. The first word is the length.
EAT_EA	The data is an ASCII name of associated data. The first word is the length.
EAT_MVMT	The data consists of two or more consecutive extended attribute values. Each value has an explicitly specified type.
EAT_MVST	The data consists of two or more consecutive extended attribute values. Each value has the same type.
EAT_ASN1	ASN.1 field data.

There are also some standard extended attribute names which are in common use. Although they all use a .Name format, this is not a requirement for extended attributes.

- .TYPE** Specifies the type of the file. Some examples are:
Plain Text
OS/2 Command File
DOS Command File
Executable
Metafile
Bitmap
Icon
and so on
- .COMMENTS** This field is intended to be a short note or reminder to the user as to what the file is about.
- .ICON** This field is used by the File or Desktop Manager to display the icon. The extended attribute data is the icon itself.
- .LONGNAME** Some file systems, such as HPFS support file names up to 254 characters long. They may also support abnormal characters, such as spaces. When a file with a long name is copied to a file system which may not support long names (FAT), applications should save the original name in this field. In this way the name is kept in case it is ever copied back to a file system which supports long names.

Extended Attributes - Usage

OS/2 itself, as well as other applications, use extended attributes. For example, OS/2 uses extended attributes to display icons, for example. The extended attribute name is .ICON and the type is EAT_ICON.

In addition, applications such as FI use extended attributes to save the information about a file without modifying the file itself. FI uses the .COMMENTS extended attribute of type EAT_ASCII to save its information.

If the .LONGNAME extended attribute is present, the OS/2 File Manager will use it to display the name of the file.

Appendix B - SEMMGR

Summary

SEMMGR is a program which manages multiple system semaphores on your behalf.

Icon



Description

SEMMGR is a detached program with which BE communicates to manage system semaphores. SEMMGR can only be run as a detached task. If you try to run it from the command line SEMMGR will report an error. If SEMMGR is not already loaded any one of the BE semaphore related commands will detach it. SEMMGR can be manually detached from the command line.

To do this, enter:

```
DETACH SEMMGR
```

from the command line.

SEMMGR creates, clears, sets, waits on and closes system semaphores. System semaphores are named objects which you may use for signalling or synchronisation between processes. The name always takes the form of:

```
\SEM\path\name.ext
```

where the path and extension (.ext) are optional. SEMMGR always uses semaphores of the form:

```
\SEM\SEMMGR\<name>
```

where <name> is the semaphore name which you supply to the BE semaphore function. This has been done to minimise the chance of interfering with other system semaphores.

The traditional use of semaphores is to control access to a resource which may not have more than one user at a time. The resource may not have more than one user because it could be damaged or behave unpredictably. For example, you would not want more than one process to access the same serial port at the same time. A semaphore can be created to represent the resource. Other processes should not access the resource unless they "own" the corresponding semaphore. Owning a semaphore is represented by setting the semaphore. When a semaphore is not owned by any one, it should be set to clear. A process should wait for a semaphore to become clear. It should then set the semaphore and continue processing.

The BE program supplies several sub functions which help you manage semaphores via SEMMGR.

- | | |
|-----------------|--|
| SEMCLEAR <NAME> | Clears a system semaphore of the name: <code>\SEM\SEMMGR\<NAME></code> . |
| SEMCLOSE <NAME> | Closes a system semaphore of the name: <code>\SEM\SEMMGR\<NAME></code> . A system semaphore can only be closed when it is cleared. This function should be used carefully as another process may be about to set the semaphore. If it has been closed and you attempt to do anything other than create it again SEMMGR will report an error. |
| SEMKILL | This command unilaterally shuts down the SEMMGR process. This should ONLY be done when there are no semaphores in use. See also SEMLIST. |
| SEMLIST | The SEMLIST command lists all currently SEMMGR managed system semaphores and their state. |
| SEMSET <NAME> | Sets a system semaphore of the name: <code>\SEM\SEMMGR\<NAME></code> . This command waits for the semaphore to be cleared before it is set again. See below for further details. |

SEMWAIT <NAME> <TIMEOUT>	Waits for a system semaphore of the name: \SEM\SEMMGR\<NAME> to become clear. If it is already clear SEMWAIT returns immediately. If no <TIMEOUT> value is specified, SEMWAIT will wait indefinitely for the semaphore to clear. If <TIMEOUT> has been specified it will wait that many milliseconds for the semaphore to clear.
---	--

Consider the following situation:

Several processes are all waiting for the same semaphore to clear. The process which owns the semaphore finishes and clears the semaphore. All of the waiting tasks find that the semaphore is clear. They all then set it, thinking that they own it, and they all then proceed to run the same resource critical task simultaneously. There can be some time between when the process finds that the semaphore is clear and when it actually sets it. To overcome this problem, SEMSET waits until semaphore is clear before it sets it and returns control back to BE.

To illustrate the use of semaphore usage, the following files have been provided.

BE-TASKS.CMD	Starts several tasks which all require access to the same critical resource.
RESOURCE.CMD	Sample of a resource critical task.
BETASK01.CMD	CMD file to wait access to RESOURCE.CMD
BETASK01.BE	BE response file for BETASK01.CMD
BETASK02.CMD	CMD file to wait access to RESOURCE.CMD
BETASK02.BE	BE response file for BETASK02.CMD
BETASK03.CMD	CMD file to wait access to RESOURCE.CMD
BETASK03.BE	BE response file for BETASK03.CMD
BETASK04.CMD	CMD file to wait access to RESOURCE.CMD
BETASK04.BE	BE response file for BETASK04.CMD
BETASK05.CMD	CMD file to wait access to RESOURCE.CMD
BETASK05.BE	BE response file for BETASK05.CMD
BETASK06.CMD	CMD file to wait access to RESOURCE.CMD
BETASK06.BE	BE response file for BETASK06.CMD
BETASK07.CMD	CMD file to wait access to RESOURCE.CMD
BETASK07.BE	BE response file for BETASK07.CMD
BETASK08.CMD	CMD file to wait access to RESOURCE.CMD
BETASK08.BE	BE response file for BETASK08.CMD

BETASK09.CMD	CMD file to wait access to RESOURCE.CMD
BETASK09.BE	BE response file for BETASK09.CMD
BETASK10.CMD	CMD file to wait access to RESOURCE.CMD
BETASK10.BE	BE response file for BETASK10.CMD

Example

```
DETACH SEMMGR
```

This is how SEMMGR can be manually detached to run as a background process. If SEMMGR has not been detached and BE needs to process semaphore functions, BE will detach it for you.

Appendix C - 2LZH.INI

2LZH.INI is a text file which enables the program 2LZH to be dynamically configured. It is a text file which can be edited by any text editor. It is used to configure the archivers and virus scanning software. If 2LZH has any problems with the information in 2LZH.INI, it will continue with its default values. The default values are the same as those specified by 2LZH.INI.

Archives Section

To configure 2LZH for a specific archive format you need to do the following:

- 1) Add a section name to the entry "Sections" under the section "[Archives]".
- 2) Add the same section name, and configure the entries "Executable", "Extension" and "Switches" as appropriate.

For example, if you wished to configure 2LZH to use ARC2, (an OS/2 version of ARC that manipulates .ARC files) you would add ARC to the "Sections" entry under the "[Archives]" section.

For example:

```
[Archives]
Sections=ARJ ZIP ZOO ARC
```

You would also add an "[ARC]" section, and then configure the "Executable", "Extensions" and "Switches" entries.

For example:

```
[ARC]
Executable=arc2.exe
Extension=.ARC
Switches=x
```

The "Executable" entry is the actual program which does the unarcng of the archives. This program must be accessible by your path statement in your CONFIG.SYS file.

The "Extension" entry is the extension of the archive. In this example, 2LZH would look for *.ARC files.

The "Switches" entry specifies the switches used by the archive program - which is specified by the "Executable" entry to actually unpack the archive.

Virus Section

The 2LZH.INI file also configures 2LZH to use virus scanning software. This is used if the -v switch is specified. 2LZH searches for a section "[Virus]" and uses the "Executable" and "Switches" entries.

The entry "Executable" is the actual program which does the virus scanning of the unpacked file. It must be accessible by your path statement as specified in your CONFIG.SYS file.

The "Switches" entry specifies the switches used by the virus scanning software.

By default, 2LZH.INI is configured to use OS2SCAN by McAfee Associates.

2LZH.INI

The supplied 2LZH.INI file looks similar to:

```
;
; 2LZH - Convert .ARJ, .ZIP and .ZOO etc to .LZH
;

[Archives]
Sections=ARJ ZIP

[ARJ]
Executable=unarj.exe
Extension=.ARJ
Switches=x

[ZIP]
Executable=unzip.exe
Extension=.ZIP
Switches=-d
```

```
; [ZOO]
; Executable=zoo.exe
; Extension=.ZOO
; Switches=x
```

```
[VIRUS]
Executable=os2scan.exe
Switches=/bell /noexpire /nopause *
```


Appendix D - WARP-INI

Summary

WARP-INI is a program used to pack the INI file used by the Windows NT versions of these programs.

Icon



Description

TM and GCD need to save some information in an INI file. The OS/2 versions of these programs use the normal OS/2 INI files. TM saves its data in the OS/2 user INI file (OS2.INI). GCD uses its own GCD.INI file. However, the OS/2 subsystem in Windows NT does not support these functions - so a custom INI file is used. All INI data is saved to C:\WARP-OS2.INI. As information is saved to and deleted from this file, the file will grow in size. The WARP-INI program packs this file by removing all deleted entries.

Example

WARP-INI

There are no switches or options for WARP-INI.

Appendix E - Extra Utilities

It order to save you time some extra public domain programs have been included. These have all been included at no extra cost to you. They have been included here to save you the effort of having to go and find them yourself. Some of the here are for use with 2LZH. However, you may wish to learn how to use them and then use them by themselves. FST which is also included, is provided as an excellent testing and reporting tool.

The files which are supplied are listed below with a brief description of each one.

It is your responsibility to unpack them and register the products (if required) as described in the licence agreements contained within the archives.

WarpSpeed Computers does not warrant or support these programs. They are provided here for your convenience only.

LH.EXE

This is the unpacked 16 bit version of the LH archiver. If you wish you may unpack LH2_222.EXE and use the 32 bit version. 2LZH requires the LZH archiving program to be called LH.EXE. Thus you would have to rename LH32.EXE to LH.EXE.

LH2_222.EXE

This is the complete self extracting archive package written by Peter Fitzsimmons. Peter has released LH into the public domain. Thus it is freely available to everyone, for whatever use, at no charge.

Peter is contactable at:

A:WARE Inc
6056 Cayeswood Court
Mississauga
Ontario L5V-1B1
Canada

E-Mail him at:

Internet sol3@olc.gvc.com

UNZ51x16.LZH

This is the 16 bit version of the Info-ZIP UNZIP program. It may be used with both OS/2 1.x and 2.x.

This version is compatible with the newer Deflation scheme which PKZIP 2.x introduced. Its versions are fully HPFS and Extended Attribute aware. This utility is publicly available, and is included at no extra cost to you. The source code for these utilities is also available for those of you who wish to view or modify it for other systems.

UARJ241P.LZH

This is version 2.41 of Robert Jung's UNARJ program. It only extracts ARJ archives which were produced by the DOS ARJ program. Currently there is no OS/2 version of ARJ. Included by kind permission of the author.

Robert is contactable at:

Robert K Jung
2606 Village Road West
Norwood,
Massachusetts 02062
USA

E-Mail him at:

Internet robjung@world.std.com
CompuServe 72077,445

OSCI251E.ZIP

This is the latest OS/2 version of SCAN and CLEAN from McAfee Associates. SCAN is a program which scans your disks and files for a large variety of viruses. CLEAN may be used to remove viruses from the infected files which SCAN finds. It is used with the -v switch of the 2LZH program.

McAfee Associates can be contacted at:

McAfee Associates
2710 Walsh Avenue, Suite 200
Santa Clara
California 95051-0963
USA

For questions, orders and problems call

Voice (408) 988-3832 (Mon-Fri, 7:00am - 5:30pm PT)

FAX (408) 970-9727 (24 h, G3-FAX):

BBS (408) 988-4004 (24 h, 32 Lines, V.32bis, USR HST-DS)

FST03F.ZIP

Fst (File System Tool) is a tool for checking HPFS and FAT drives, like CHKDSK, and for information about HPFS and FAT drives. Unlike CHKDSK, fst does not attempt to fix the errors it finds. Instead, fst prints the relevant sector numbers for each defect so that you can examine and/or fix the defects with DISKEDIT.

Fst is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2, or (at your option) any later version.

Eberhard Mattes wrote fst. E-Mail him at:

Internet mattes@azu.informatik.uni-stuttgart.de

Appendix F - CompuServe

WarpSpeed computers is pleased to announce that it now offers support through the Other Vendors section (section 1) of the OS2AVEN forum. Simply enter:

GO OS2AVEN

at any ! prompt. Or you can EMail me directly: 100250,1645.

If you are not a member of CompuServe and wish to join, please contact the customer service centre in your country.

CUSTOMER SERVICE GENERAL INFORMATION

Local Customer Service is available from around the world. You may reach Customer Service by sending electronic mail messages via the Online Feedback system, by telephone, by postal mail or by fax.

You can leave an electronic message for Customer Service any time - day or night - in the Online Feedback area. To access Feedback: select "Feedback" from the previous menu, or GO FEEDBACK at any point online. The feedback area is part of basic services, and is also free of connect time charges for members on the alternative pricing plan. Applicable network charges (like FALNET from Australia or New Zealand) still apply.

You may also send messages via Fax. To ensure prompt delivery of a faxed letter, be sure to include a cover page clearly stating your name, User ID number, and the name of the department (and/or person) to whom the letter is being sent.

ARGENTINA CUSTOMER SERVICE

Mailing Address:

CompuServe S.A. Argentina
Mr. Carlos Garbesi / Mr. Gabriel Saiz
Av. Rivadavia 969 2 do.Piso Frente
Buenos Aires 1002
Argentina

Phones:

Inside Argentina:	(01) 345-3817, (01) 345-3871 (01) 345-0836, (01) 345-3694
Outside Argentina:	(+541) 345-3817, (+541) 345-3871 (+541) 345-0836, (+541) 345-3694

Fax:

Inside Argentina:	(01) 345-0825
Outside Argentina:	(+541) 345-0825

Hours:

Monday - Friday:	9:00 - 17:00
------------------	--------------

AUSTRALIA AND NEW ZEALAND CUSTOMER SERVICE

Mailing Address:

CompuServe Pacific
Fujitsu Australia Ltd
475 Victoria Avenue
Chatswood, NSW
Australia

Phones:

Inside Australia:	(02) 9410 4223
Outside Australia:	(+612) 9410 4223
Freephone in Australia:	008 025 240
Freephone in New Zealand:	0800 446 113

Fax:

Inside Australia:	(02) 9410 4223
Outside Australia:	(+612) 9410 4223

Hours:

Monday - Friday: 7:00 - 19:00 (EST)
9:00 - 21:00 (New Zealand Time)

CANADA CUSTOMER SERVICE

Mailing Address:

CompuServe
Attn: Customer Service
P.O. Box 20212
5000 Arlington Centre Blvd.
Columbus, Ohio 43220 USA

Phones:

Inside Canada: (614) 457 8650
Outside Canada: (+1 614) 457 8650

Fax:

Inside Canada: (614) 457 8149
Outside Canada: (+1 614) 457 8149

Hours:

Monday - Friday: 8:00 - 24:00 (EST)
Saturday - Sunday: 12:00 - 22:00 (EST)

CHILE CUSTOMER SERVICE

Mailing Address:

CHILEPAC S.A.
Gerencia Red de Datos
Morande 147
Santiago
Chile

Phones:

Inside Chile: (02) 696 8807
Outside Chile: (+56 2) 696 8807
Or dial your local
ChilePac or Telex-Chile 294400 CHPAC CL

Fax:

Inside Chile: (02) 698 1474
Outside Chile: (+56 2) 698 1474

Hours:
Monday - Friday: 9:00 - 17:00

FRANCE CUSTOMER SERVICE

Mailing Address:

CompuServe Information Service SARL
Centre ATRIA
Rueil 2000 - 21, Av Edouard Belin
92566 RUEIL MALMAISON CEDEX
France

Phones:

Sales: AZUR phone number
within France 36 63 81 22
Service: AZUR phone number
within France 36 63 81 31
Outside France: (+33 1) 47 14 21 60
Fax: Outside France: (+33 1) 47 14 21 51
Hours: Monday - Friday: 8:30 - 19:30

GERMANY CUSTOMER SERVICE

Mailing Address:

CompuServe GmbH
Postfach 11 69
D-820001 Unterhaching bei
Muenchen
Germany

Phones:

Sales: Freephone in Germany: 0130 37 32
Freephone in
Switzerland: 55 31 79
Outside Germany: (+49 89) 66 535 111
Service: Freephone in Germany: 0130 86 4643
Outside Germany: (+49 89) 66 535 222

Fax:
Sales: Outside Germany: (+49 89) 55 535 242
Service: Outside Germany: (+49 89) 66 535 241
Hours: Monday - Friday: 9:00 - 20:00

HONG KONG CUSTOMER SERVICE

Mailing Address:

CompuServe Hong Kong
Hutchison Information Services, Ltd
30th Floor, One Pacific Place
88 Queensway
Hong Kong

Phones:
Sales: Inside Hong Kong: 867 0118
Outside Hong Kong: (+852) 867 0118
Service: Inside Hong Kong: 867 0102
Outside Hong Kong: (+852) 867 0102
Fax:
Inside Hong Kong: 877 4523
Outside Hong Kong: (+852) 877 4523
Hours: Monday - Friday: 9:00 - 18:00
Saturday: 9:00 - 12:00
Closed Sundays and
Holidays.

HUNGARY CUSTOMER SERVICE

Mailing Address:

Mircosystem R.
1122 Budapest Xii,
Varosmajor u. 74
Budapest
Hungary

Phones:

Sales:	Inside Hungary:	156 5366
	Outside Hungary:	(+36 1) 156 5366
Service:	Inside Hungary:	156 5366
	Outside Hungary:	(+36 1) 156 5366
Fax:	Inside Hungary:	155 9296
	Outside Hungary:	(+36 1) 155 9296
Hours:	Monday - Friday:	8:30 - 16:30

ISRAEL CUSTOMER SERVICE

Mailing Address:

CompuServe Israel
Trendline Information and Communications Services, Ltd
22 Yavne Street
Tel-Aviv
Israel

Phones:

Sales:	Inside Israel:	(03) 525 1948
	Outside Israel:	(+972 3) 525 1948
Service:	Inside Israel:	(03) 525 1946, (03) 525 2444 (03) 290 466
	Outside Israel:	(+972 3) 525 1946, (+972 3)2444 (+972 3) 290 466

Fax:

Inside Israel:	(03) 200419
Outside Israel:	(+972 3) 200419

Hours:

Sunday - Thursday:	9:00 - 17:00
--------------------	--------------

JAPAN CUSTOMER SERVICE

Mailing Address:

NIFTY Corporation
8th Floor, Omori Bellport A,
Minami-oi 6-26-1, Shinagwa-ku
Tokyo 140

Japan

Phones:

Tollfree Inside Japan: 0120 22 1200
Outside Japan: (+81 3) 5471 5806

Fax:

(+81) 3 5471 5890, (+81) 3 5471 5891

Hours:

Monday - Friday: 9:00 - 19:00 (JST)
Saturdays: 9:00 - 17:50 (JST)

KOREA CUSTOMER SERVICE

Mailing Address:

POSDATA CO., LTD.
Overseas Information Services
2F, Taehanjedang Building
7-23, Shinchun-dong, Songpa-ku
Seoul
Korea

Phones:

Tollfree Inside Korea: 080 022 7400
Outside Korea: (+82 2) 441 1327

Fax:

(+82 2) 411 1555

Hours:

Monday - Friday: 9:00 - 19:00
Saturdays: 9:00 - 17:50

SOUTH AFRICA CUSTOMER SERVICE

Mailing Address:

CompuServe Africa
P.O. Box 72668
Lynnwood Ridge
South Africa 00400

Phones:

Sales: Freephone Inside SA: 0800 112252
Outside SA: (+27 12) 841 2538
Service: Inside SA: (012) 841 2530
Outside SA: (+27 12) 841 2530
Fax: (012) 841 3604
Hours: Monday - Friday: 9:00 - 17:00 (RSA) Standard Time

TAIWAN CUSTOMER SERVICE

Mailing Address:

TTN-Serve
Taiwan Telecommunications Network Svcs. Co., Ltd.
Far East ABC Intelligent Science Park
1st Floor, No. 13, Lane 50, Nan-Kang Road, Section 3,
Taipei
Taiwan R.O.C.

Phones: Freephone inside 080 251 009
Taiwan:
Outside Taiwan: (+886 2) 651 6899
Fax: (+886 2) 651 1801
Hours: Monday - Friday: 9:00 - 18:00
Saturdays 9:00 - 12:00

UNITED KINGDOM CUSTOMER SERVICE

Mailing Address:

CompuServe Information Service (UK) Ltd.
1 Redcliff Street
P.O. Box 676
Bristol BS99 1YN
United Kingdom

Phones:

Sales: 0800 289 378
Service: Inside the UK: 0800 289 458
Outside the UK: (+44) 272 760 680
Fax: (+44) 272 252 210
Hours: Monday - Friday: 9:00 - 21:00

UNITED STATES CUSTOMER SERVICE

Mailing Address:

CompuServe
Attn: Customer Service
P.O. Box 20212
5000 Arlington Centre Blvd.
Columbus, Ohio 43220
USA

Phones:

Freephone inside the US: (800) 848 8990
Includes Puerto Rico, St.
Thomas, St. John and St.
Croix.

Outside the US: (+1 614) 529 1340

Fax:

Inside the US: (614) 457 8149

Outside the US: (+1 614) 457 8149

Hours:

Monday - Friday: 8:00 - 24:00 (EST)

Saturday - Sunday: 12:00 - 22:00 (EST)

VENEZUELA CUSTOMER SERVICE

Mailing Address:

CompuServe C.A. Venezuela
Plaza Venezeuela - Torres Capriles
Piso 4, Oficina 401
Caracas
Venezeula

Phones:

Inside Venezuela: 02 793 2984, 02 793 2384, 02 791 8694
Outside Venezuela: (+58 2) 793 2984, (+58 2) 793 2984
(+58 2) 791 8694

Fax:

Inside Venezuela: 02 793 1952
Outside the Venezuela: (+58 2) 793 1952

Hours:

Monday - Friday: 8:00 - 17:00

Contacting Customer Service

If there is no local Customer Service listing in your country, please try contacting the Customer Service office nearest you or call CompuServe Customer Service in the United States for further assistance.

Appendix G - HPFS Internals

Background Information

The High Performance File System (HPFS) was introduced with OS/2 version 1.2. It was designed by Gordon Letwin, the chief architect of the OS/2 1.x operating systems. HPFS uses different structures, caching and allocation techniques to achieve superior performance over the File Allocation Table (FAT) file system.

Base Structures

What follows is a highly technical discussion of the internals of the HPFS disk structures and organisation.

Boot Blocks

The HPFS has only three items which have a fixed location. The first is the volume boot sector which starts at logical sector 0. Sectors 0 to 15 are reserved for the boot sector and up to 8K of disk bootstrap code. The boot code can access HPFS in a restricted mode; enough to ensure the system is booted and running to the point of being able to load HPFS.IFS - which is the real HPFS driver.

BOOT BLOCKS

BIOS Parameter Block Volume Name 32 bit Volume ID Bootstrap Program (1-16 sectors)

Super Block

Located at logical sector 16 is the Super Block. The Super Block contains the following information:

Super Block

Pointer to Root Directory FNode
Size of volume
Number of Bad sectors
Pointer to Bitmap sectors lists
Pointer to Bad sector lists
Date and Time of last CHKDSK /F
Date and Time of last optimise
Size of directory band
Start and end of directory band
Pointer to Bitmap for directory band
Pointer to ACL information

Super Block Information:

Root Directory FNode	This is the logical sector number of Root Directory FNode.
Size of volume	The size of the volume in logical sectors.
Number of Bad sectors	The total number of bad sectors on the disk.

Bitmap sector lists	This is the sector number of a group of four sectors which form a list of all of the bitmap sectors. Each group of bitmap sectors is four sectors long. The four sectors of each bitmap represents 16,384 bits. Each bit represents the usage of one sector. A 0 means the sector is used and a 1 means that the sector is free. 16,384 sectors is 8Mb, and thus the size of the bands on a HPFS disk. This is an arbitrary limit. However, it may change in future revisions of the HPFS. To maximise the amount of contiguous free space two 8Mb bands are grouped back to back, giving a maximum of 16Mb of contiguous file space. Because the usage of the disk is mapped out by the bitmaps, it allows the HPFS to minimise file fragmentation when a file is created.
Bad sector lists	This is the sector number of a group of four sectors which form the list of currently marked bad sectors. The first four bytes of the list is the sector number of the next bad sector list (zero if there are no more). This leaves room for 511 bad sectors. Bad sectors are marked as used in the bitmaps.
Last CHKDSK /F	This is the date and time at which the last CHKDSK /F was run. GMT is used. The value is zero if CHKDSK has never been run.
Last optimise	This is the date and time of when the last disk optimise was run. GMT is used. The value is zero if the disk has not been optimised.
Size of directory band	The number of sectors which form the directory band. The directory band is located near the seek centre of the disk.
Start, end of dir. band	The starting and ending sector of the directory band.
Bitmap for dir. band	The sector number of the bitmap particularly for the directory band.

ACL Information There are eight sectors which are allocated near the directory band which is used by HPFS386 to store ACL (Access Control List) information.

The Super Block is only modified by disk maintenance utilities such as CHKDSK HPFS-Bad, HPFSDfrg and HPFSView.

A complete description of all fields in the Super Block is contained in Appendix I - DiskEdit Modules.

Spare Block

The Spare Block is at logical sector 17. It contains the following information:

Spare Block

Volume status flag Pointer to Hotfix lists Number of Hotfixes Number of Hotfixes in use Number of spare DIRBLKs Pointer to Code Page Information Number of Code Pages Pointers to spare DIRBLKs
--

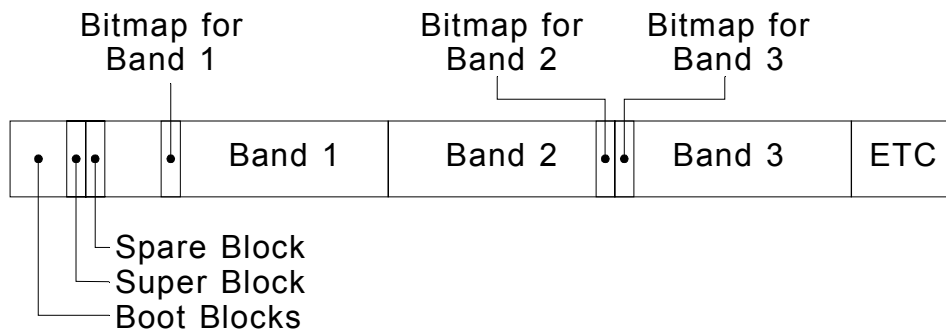
Volume status flag The partition status flag reflects the current status of the partition. It also includes the dirty flag which tells the system whether the volume was shut down properly (clean) or if it was improperly stopped and there was information in memory which was still waiting to be written to disk (dirty). When the system boots it checks the state of the dirty flag. If it is dirty, CHKDSK is run to resolve any differences, and if necessary rebuild the necessary disk structures. If the boot volume is dirty the boot procedure stops and you must boot OS/2 from a floppy disk to resolve the problems.

Hotfix list	This is the sector number of the hot fix list - which is four sectors long. This list contains the sector numbers of the sectors which have gone bad and the sector numbers of the good sectors to which they have been remapped. The HPFS checks any real sector reads/writes (as opposed to reads/writes from cached sectors) with this list and substitutes them as necessary.
Number of hotfixes	This is the total number of hotfixes which the volume has.
Number of hotfixes in use	This is the number of hotfixes which are currently used.
Number of spare DIRBLKs	The system maintains a number (generally 20) of spare DIRBLKs. They are usually at the end of the directory band as defined by the Super Block.
Code Page Information	The sector number of the Code Page Information sector for the volume.
Number of Code Pages	The number of Code Pages which are on the volume.
Pointers to spare DIRBLKs	This is a list of the sector numbers of the spare directory DIRBLKs which the system maintains.

The Spare Block is modified, (although infrequently), as the system operates.

A complete description of all fields in the Spare Block is contained in Appendix I - DiskEdit Modules.

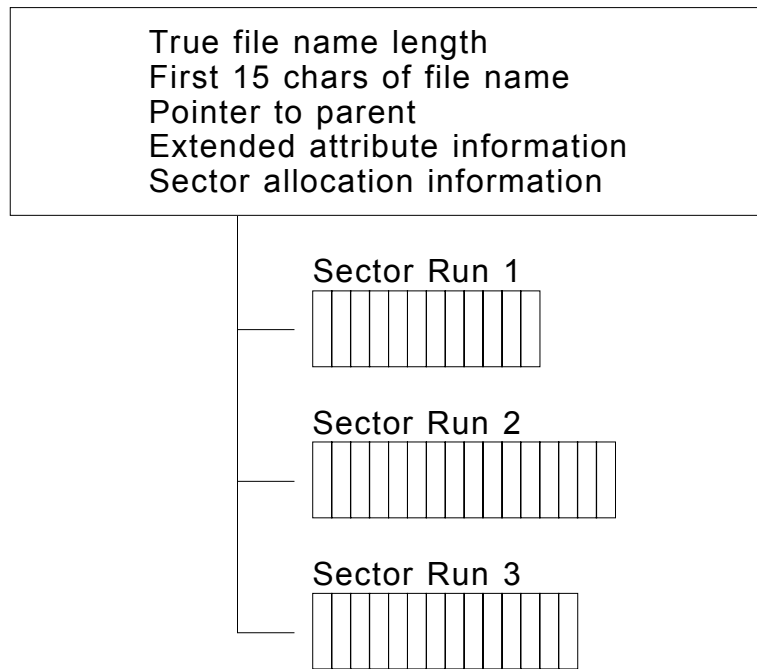
As can be inferred from the above information, the layout of the HPFS volume looks similar to the diagram below:



FNodes

The fundamental file system object on an HPFS volume is an FNode (pronounced "eff node"). The FNode contains the true length of the file name, the first fifteen characters of the file name (so that disk recovery utilities such as HPFS-UD can recover deleted files or files from damaged volumes), a pointer to the FNode's parent directory (to enable you to build up path information), extended attribute information and information regarding the position of the data sectors for the file itself. If the file is in eight or less extents, the FNode has room to contain the starting sector and the number of sectors for each sector run.

FNode



DIRBLKs

Similar to files, directories are anchored on FNodes. A pointer to the root directory FNode is located in the Super Block. Directories are built up of DIRBLKs. Each DIRBLK is four sectors long. They are allocated as four contiguous sectors. Once the directory band is full, they are allocated wherever there is space. The directory band is located near the seek centre of the disk for performance reasons (less head movement). The DIRBLKs are composed of variable length records which represent each directory entry. They are called DIRENTs. Each directory entry contains the following information:

Directory Entry

Size of entry
Flags for file
Attribute of file
Pointer to file FNode
Date and Time of last write
Date and Time of last access
Date and Time of creation
Length of file
Size of extended attributes
Code Page of file name
Length of the file name
Complete file name

When a file is deleted the complete directory entry is overwritten. Thus, information such as the complete filename and other attributes is lost. The undelete process can only process information found in deleted FNodes; which is why only a limited amount of information about files can be recovered from HPFS volumes.

Extended Attributes and ACLs

There is room for 316 bytes of extended attribute data, or ACL information in the file's FNode. If the amount of extended attribute data exceeds this it can be moved to external sector runs which contain the data; or even an ALSEC to describe heavily fragmented extended attribute data.

There is a bug in the current implementations (both HPFS and HPFS386) of the HPFS.IFS driver. A file's EA's will be lost when the number of extents needed to store the EA exceeds 40.

This is scheduled to be fixed in a future release.

Appendix H - DESVR

Summary

DESVR is a program which runs on remote machines to enable you to perform the remote disk accessing functions.

Icon



Description

DESVR can only run as a detached process. To detach DESVR, from the command line enter:

```
DETACH DESVR
```

on any remote machine which you wish to access remotely. This enables the following programs to access remote disk drives:

- DiskEdit
- DS
- FAT-Bad
- FAT-Dfrg
- FAT-Ext
- FAT-Info
- FAT-Null
- FAT-UD
- FAT-View
- HPFS-Bad
- HPFS-Ext
- HPFS-UD
- HPFSDfrg
- HPFSInfo
- HPFSNull
- HPFSStat
- HPFSView

To access the remote drives the above programs support the following command line syntax:

```
<Program> -on \\<Remote Machine Name>
```

Where <Remote Machine Name> is the name of the remote server or workstation. The Remote Machine Name will have been set when the network software was installed.

WARNING:

DESVR bypasses any and all security provided by the network. You may not wish to run it on all machines on your network.

Any file specifications or drive specifications refer to those on the remote machine (unless specifically otherwise noted) - not those on your local machine. For example:

```
DISKEDIT C: -on \\MAIN
```

DiskEdit will attempt to access drive C: on the remote machine named \\MAIN.

If the machine name does not exist you will see this error message:

```
SYS0053: The network path was not found.
```

If DESVR is not running on the remote machine you will receive the following error message:

```
SYS0003: The system cannot find the path specified.
```

Appendix I - DiskEdit Modules

Summary

The disk editor (DiskEdit.Exe) comes with a number of modules which allow you to edit disks, both logical and physical, both local and remote, in a variety of ways.

DiskEdit itself only manages the higher level functions. The real editing functionality is contained within the DE*.DLL's themselves. Each DE DLL has a variety of editing modes. Not every DLL supports every editing mode. The available editing modes and the DLL's which support them are as follows:

Edit Mode	DISPLAY	PHYS	RAW	HPFS	FAT	PIPE
Edit Partition Table	PART TABL	•	•	•	•	•
Edit Boot Sector	BOOT SECT		•	•	•	•
Edit FAT	FAT				•	
Edit Directory	DIR				•	
Edit Raw	RAW	•	•	•	•	•
Edit Super Block	SUPER BLK			•		
Edit Spare Block	SPARE BLK			•		
Edit Bitmap	BITMAP			•		
Edit Directory Bitmap	DIR BITMP			•		
Edit DIRBLK	DIRBLK			•		
Edit FNODE	FNODE			•		
Edit ALSEC	ALSEC			•		
Edit Code Page Information	CODE PG I			•		
Edit Code Page	CODE PG			•		
Edit Bad Block List	BAD BLK L			•		
Edit Hot Fix List	HOT FIX L			•		
Edit Bitmap List	BITMAP L			•		

Edit Partition Table

The Partition Table editing mode allows you to edit the partition table for the selected drive. It is only available when you are editing the very first sector on a hard disk.

When this editing mode is active you will see a window similar to the following:

00	05	131	0	1	1023	63	32	268288	2408448
80	0a	130	0	1	130	63	32	266240	2048
00	07	30	0	1	129	63	32	61440	204800
00	14	0	1	1	29	63	32	32	61408

As you select each column to edit you will be given a help message below the editing window.

The ten columns are:

Column One - Partition Status

This value indicates the current partition status. It is a hexadecimal number. A value of 00 indicates that the partition is not bootable. A value of 80 indicates that the partition is bootable.

Column Two - Partition Type

This value defines the type of the partition. It is a hexadecimal number. The following table lists some of the values which you may find. It is by no means exhaustive.

Partition	Description
00	Unused Partition
01	DOS, 12-bit FAT
02	XENIX System, includes SCO/XENIX
03	XENIX User, includes SCO/XENIX
04	DOS, 16-bit FAT
05	DOS and OS/2, >32MB support. Defines an Extended partition which may include other partition types.
06	DOS, >32MB support, up to 64K Allocation unit
07	OS/2, >32MB partition support (IFS) [HPFS]
08	AIX
08	OS/2 (through Version 1.3 only)
08	DELL partition spanning multiple drives (array)
08	Commodore DOS Partition
09	AIX
0A	OS/2 Boot Manager Partition
0B - 0D	Available for assignment

0E - 0F	Reserved
10	Reserved
11	OS/2 Boot Manager: DOS - Inactive type 1
12	Reserved
13	Available for assignment
14	OS/2 Boot Manager: DOS - Inactive type 4
15	Available for assignment
16	OS/2 Boot Manager: DOS - Inactive type 6
17	OS/2 Boot Manager: DOS - Inactive type 7
18 - 20	Available for assignment
21	Reserved
22	Available for assignment
23 - 24	Reserved
25	Available for assignment
26	Reserved
27 - 30	Available for assignment
31	Reserved
32	Available for assignment
33 - 34	Reserved
35	Available for assignment
36	Reserved
37 - 3F	Available for assignment
40	Series/1 Disk
41	Personal RISC Boot Partition
42 - 4F	Available for assignment
50	OnTrack Disk Manager
51	OnTrack Disk Manager
52	Reserved
53 - 55	Available for assignment
56	Reserved
57 - 60	Available for assignment
61	Reserved
62	Available for assignment
63	SCO UNIX
64	Novell
64	Speedstore
65	Novell 286 Netware
66	Novell 386 Netware
67	Novell (future use)
68	Novell (future use)
69	Novell (future use)
6A - 70	Available for assignment

71	Reserved
72	Available for assignment
73 - 74	Reserved
75	PC/IX
76	Reserved
77 - 79	Available for assignment
80 - 81	Reserved
82	Prime or Linux
83	Apple Computer or Linux Swapper Partition
84	System Hibernation for APM 1.1
85 - 85	Available for assignment
86	Reserved
87	HPFS FT mirrored partition
88 - 92	Available for assignment
93 - 94	Reserved
95 - A0	Available for assignment
A1	Reserved
A2	Available for assignment
A3 - A4	Reserved
A5	Available for assignment
A6	Reserved
A7 - B0	Available for assignment
B1	Reserved
B2	Available for assignment
B3 - B4	Reserved
B5	Available for assignment
B6 - B8	Reserved
B9 - C0	Available for assignment
C1	Reserved
C2 - C3	Available for assignment
C4	Reserved
C5	Available for assignment
C6	Reserved
C7	HPFS FT disabled mirrored partition
C8 - D7	Available for assignment
D8	CP/M 86
D9 - DA	Available for assignment
DB	Reserved
DC - E0	Available for assignment
E1	Speedstore
E2	Available for assignment
E3	Storage Dimensions (Maxtor Retail Subsidiary)

E4	Speedstore
E5 - E6	Reserved
E7 - F0	Available for assignment
F1	Storage Dimensions (Maxtor Retail subsidiary)
F2 - F3	Reserved
F4	Storage Dimensions (Maxtor Retail subsidiary)
F5	Available for assignment
F6	Reserved
F7 - FD	Available for assignment
FE	IBM PS/2 IML
FF	Bad Block Tables - Must be on cylinder 0

Column Three - Beginning Cylinder

This value is the cylinder which marks the beginning of the partition.

Column Four - Beginning Head

This value is the head which marks the beginning of the partition.

Column Five - Beginning Sector

This value is the sector which marks the beginning of the partition.

Column Six - Ending Cylinder

This value is the ending cylinder of the partition.

Column Seven - Ending Head

This value is the ending head of the partition.

Column Eight - Ending Sector

This value is the ending sector of the partition.

Column Nine - Partition Offset

This value is the offset in sectors to the beginning of the partition relative to the beginning of this logical drive.

Column Ten - Partition Size

This is the size of the partition in sectors. Each sector is generally 512 bytes long.

Edit Boot Sector

The Boot Sector editing mode allows you to edit the boot sector for the selected drive. The Bios Parameter Block (BPB) is contained in the boot sector. It is this information which an operating system reads to determine the configuration of the disk. It is only available when you are editing the very first sector of a floppy disk (typically cylinder 0, head 1, sector 0 of a hard disk).

See also: DI.

The fields which are defined are:

.JMP

These three bytes are machine code instructions which are a jump to the boot code for this disk. This field is a hexadecimal string.

.OEM

These 8 bytes are reserved for the Original Equipment Manufacturer. This field is an ASCII string.

.usBytesPerSector

Sectors are the smallest amount of data which a disk can either read or write. It is determined when the disk is physically formatted. It is usually 512, but it can be 128, 256, 1024 or 2048 as well. This field is a decimal number.

.bSectorsPerCluster

OS/2 and DOS allocate sectors in groups called "Clusters". This is the minimum number of disk sectors which DOS or OS/2 can allocate to a file. This value varies. It depends on the size and type of media. This field is a decimal number.

.usReservedSectors

This is the number of sectors reserved for use by the operating system. This field is a decimal number.

.cFATs

The File Allocation Table (FAT) file system uses tables to save information about where files reside. To assist in maintaining file system integrity more than one copy may be kept. The default is 2. This field is a decimal number.

.cRootEntries

The FAT file system requires a fixed space to be reserved for the root directory. The size of this space is determined when the disk is formatted. The value varies as it depends on the size and type of media. This field is a decimal number.

.cSectors

This specifies the number of sectors available on the disk. This value is only valid for partitions of less than 32Mb in size. This field is a decimal number.

See also: `.cLargeSectors`.

.bMedia

Specifies a hexadecimal number which lists the type of media in the disk drive. This field is a hexadecimal number. Some common values are:

F0	A 1.44Mb 3.5" floppy disk
F8	A hard disk.
F9	A 1.2Mb 5.25" floppy disk OR A 720K 3.5" floppy disk.
FD	A 360K 5.25" floppy disk.
FE	A 160K 5.25" floppy disk.
FF	A 320K 5.25" floppy disk.

.usSectorsPerFAT

Specifies the number of sectors occupied by each FAT in the FAT file system. This field is a decimal number.

.usSectorsPerTrack

Each track is divided up into a number of sectors. This is dependent on the disk drive itself. This field is a decimal number.

.cHeads

Specifies the number of heads (or sides) of the logical disk. This field is a decimal number.

.cHiddenSectors

Under FAT file systems this specifies the number of sectors reserved for system use. This field is a decimal number.

.cLargeSectors

This is the number of sectors available on a large partition. A large partition is > 32Mb in size. This field is a decimal number.

See also: .cSectors.

.bDriveNo

This is the drive number. The drive number is used internally by DOS. This field is a decimal number.

.bReserved

Reserved. This field is a decimal number.

.bExtendedSig

This value, if it is 41 (for FAT) or 40 (for HPFS), specifies that this is an extended boot record. An extended boot record has the .ulVolID, .bVolLabel and .bVolType fields in it. This field is a hexadecimal number.

.ulVolID

This is the 32 bit volume ID for this disk. This field is a hexadecimal number.

.bVolLabel

This is the 11 character Volume Label. It is set when the disk is formatted. This field is an ASCII string.

.bVolType

These 8 characters specify the volume (or file system) type. This field is an ASCII string.

Edit FAT

The FAT editing mode enables you to edit the File Allocation Table of a FAT disk. It is this table which specifies the actual disk layout of files on a FAT disk. The FAT is a simple chain of numbers. Each number representing a cluster of data. We follow this chain until we reach the end of the chain - which is marked as <EOF>. The other values of note in the FAT are zero (0), which means that the specified cluster is free or unused and <BAD> which indicates that the particular cluster is unsuitable for use.

The following paragraph describes how the FAT works. A file must have a directory entry in either the root directory or a sub directory cluster. A sub directory is simply a special file (allocated and used like any other) except that the file information in the file is actually a set of directory entries. Each of these directory entries has multiple fields in it. One of which is the starting cluster. If the file size is non-zero, then the starting cluster number is the first cluster of data for that file. The next cluster is represented in the FAT. Take the example below: If a file has a starting cluster of two (2), then the file only contains one cluster's worth of data. The chain ends at cluster 2, as indicated by the <EOF>. If the starting cluster of a file is nine (9), then the file's data is contained in clusters, 9, 10, 11 all the way through to cluster number 93.

Note:

The F12 key may be used at any time to jump to the Current Cluster - as displayed in the top right hand corner of the window.

FAT Disk D:	LSN 1	Cl 0000	Hd 001	Sect 002	FAT 1	Current	Cluster	2			
<EOF>	4	10547	<EOF>	7	8	<EOF>	10	11	12	13	14
15	16	17	18	19	20	21	22	23	24	25	26
27	28	29	30	31	32	33	34	35	36	37	38
39	40	41	42	43	44	45	46	47	48	49	50
51	52	53	54	55	56	57	58	59	60	61	62
63	64	65	66	67	68	69	70	71	72	73	74
75	76	77	78	79	80	81	82	83	84	85	86
87	88	89	90	91	92	93	<EOF>	95	96	97	98
99	100	101	102	103	104	105	106	107	108	109	110
111	112	113	114	115	116	117	118	119	120	121	122
123	124	125	126	127	128	129	130	131	132	133	134
135	136	137	138	139	140	141	142	143	144	145	146
147	148	149	150	151	152	153	154	155	156	157	158
159	160	161	162	163	164	<EOF>	<EOF>	<EOF>	<EOF>	169	170
171	172	173	174	175	176	177	178	179	180	181	182
183	184	185	186	187	188	189	190	191	192	193	194
195	196	197	198	199	200	201	202	203	204	205	206
207	208	209	210	211	212	213	214	215	216	217	218
219	220	221	222	223	224	225	226	227	228	229	230
231	232	233	234	235	236	237	238	239	240	241	242
243	244	245	246	247	248	249	250	251	252	253	254
255	256	257	258	259	260	261	262	263	264	265	266
267	268	269	270	271	272	273	274	275	276	277	278
279	280	281	282	283	284	285	286	287	288	289	290
291	292	293	294	295	296	297	298	299	300	301	302
303	304	305	306	307	308	309	310	311	312	313	314
315	316	317	318	319	320	321	322	323	324	325	326
327	328	329	330	331	332	333	334	335	336	337	338
339	340	341	342	343	344	345	346	347	348	349	350
351	352	353	354	355	356	357	358	359	360	361	362
363	364	365	366	367	368	369	370	371	372	373	374

Press B for <BAD>, E for <EOF> or numeric for cluster #, J for Jump to Cluster
Press F1 for Help

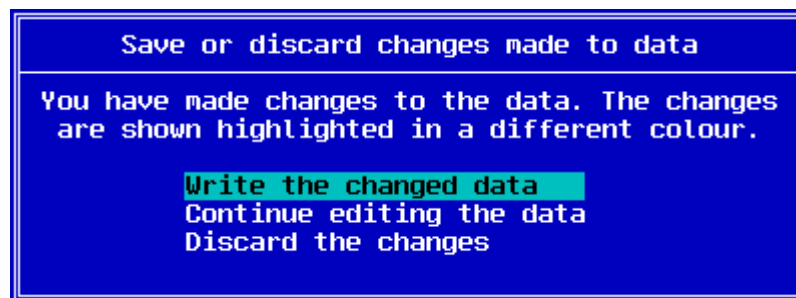
You may use the arrow keys to move around the FAT to edit or change values. The FAT can be very large. On a 2Gb disk there are 65,506 clusters. This can make scrolling through them very difficult. To ease this situation, you can press the J key and you will be prompted with this dialog box:

Jump to Cluster Number
Enter a cluster to go to.

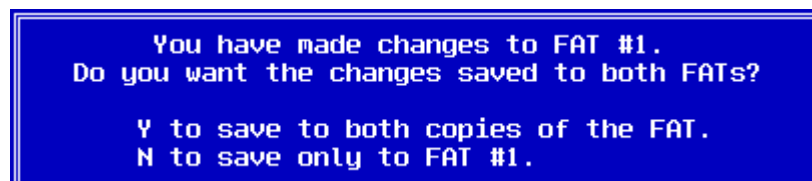
Cluster : 2

Simply enter the number that you wish to go to and you will be taken there.

Once you have finished making your changes, you will be prompted to save them. At this point you will be presented with this dialog box:



Once you have chosen to write your changes back to disk you will be prompted with another dialog box. Generally there are two copies of the FAT for data redundancy purposes. It is important to keep them both the same. Thus at this point you are prompted to make changes to both or just one of them.



Edit Directory

The Edit Directory mode allows you to edit FAT directories in their native format. The following columns are available for editing.

Filename	Ext	File Size	Date	Time	Start Clust	EA Index	Attributes								
							R	A	S	H	D	U	V		
AUTOEXEC	BAT	622	02-03-1995	15:20:58	218	0	A								
BRIDGE	BMP	21518	19-03-1995	15:07:40	219	0	A								
LOVELY	BMP	690462	19-06-1994	17:07:32	220	0	A								
AUTOTRAX	LZH	576239	03-10-1995	12:17:18	242	0	A								
2GB DISK		0	08-08-1996	15:42:16	0	0									U
BOOT	CFG	1532	20-09-1995	16:46:18	261	0	A								
SD	INI	2497	06-10-1995	15:44:32	262	0	A								
HDP2INST	MIM	2715326	11-04-1996	22:04:36	263	0	A								
HDP2INST	UUE	2759664	06-07-1996	09:24:24	346	0	A								
WP ROOT	SF	268	11-05-1995	09:44:40	217	1	A	S	H						
EA DATA	SF	65536	01-08-1996	12:03:34	215	0	R	A	S	H					
LOVELY	LZH	590193	22-04-1994	16:26:50	431	0	A								
DEVTECH3	DOC	32002	03-04-1996	12:56:30	450	0	A								
σDP2INST	EXE	2002639	31-03-1996	11:58:04	451	0	A								
A&L		0	01-08-1996	12:03:42	513	0									D
VACPP	PKG	75156	17-04-1996	14:02:14	514	0	A								

FileName

This is the filename. It is 8 characters wide.

Ext

This is the filename's extension. It is 3 characters wide.

File Size

This is the size of the file in bytes.

Date

This is the date (in Day-Month-Year format) on which the file was last modified.

Time

This is the time (in HH:MM:TwoSec format) on which the file was last modified. The TwoSec field is the time to the nearest two seconds.

Start Clust

This is the starting cluster for the file's data.

EA Index

This is the Extended Attribute index into the "EA DATA . SF" file (the Extended Attribute holding file for FAT partitions).

Attributes

This column has the following sub headings:

R	This is the READ-ONLY attribute bit. It indicates that the file can not be written to or changed.
A	This is the ARCHIVE attribute bit. It indicates that the file has been changed since the last backup was performed and that it should be backed up.
S	This is the SYSTEM attribute bit. It indicates that the file is a system file and thus should not be modified or moved. It is normally used with the HIDDEN attribute bit.
H	This is the HIDDEN attribute bit. It indicates to the system that this directory entry should not be shown in normal directory searches.
D	This is the DIRECTORY attribute bit. This indicates that the directory entry is a subdirectory.
V	This is the VOLUME label attribute bit. This is the volume label for the disk. There should only be one directory entry with this bit set, and it must be in the root directory.

Note:

The F12 key can be used to jump to the Start Cluster of the currently selected directory entry.

Edit Raw Hex

The edit Raw Hexadecimal mode is provided across all of the DE DLL's. This is the default editing mode. It allows you to edit data in both hexadecimal and ASCII.

Edit Super Block

The edit Super Block function allows you to edit the Super Block of HPFS disks in their native format. The fields which define the SUPERB structure are as follows:

.sig1

This is the first signature value for the Super Block. It should be F995E849.

.sig2

This is the second signature value for the Super Block. It should be FA53E9C5.

.bVersion

This is the version number of the HPFS File System structures. It is normally 2.

.bFuncVersion

This is the functional version number. This is the oldest version of the file system which can understand this disk. It is normally 2.

.usDummy

This is a dummy field used to align everything to a 32 bit boundary.

.lsnRootFNode

This is the Logical Sector Number of the root directory FNODE.

.culSectsOnVol

This is the total number of sectors on the volume.

.culNumBadSects

This is the number of bad sectors on the volume.

.rspBitMapIndBlk.lsnMain

This is the Logical Sector Number of the starting sector of the Bitmap lists.

.rspBitMapIndBlk.lsnSpare

This is unused.

.rspBadBlkList.lsnMain

This is the Logical Sector Number of the starting sector of the Bad Block lists.

.rspBadBlkList.lsnSpare

This is unused.

.datLastChkdsk

This is the date and time of when CHKDSK /F was last run.

.datLastOptimize

This is the date and time of the last disk optimise.

.clsnDirBlkBand

This is the number of sectors in the DIRBLK band.

.lsnFirstDirBlk

This is the Logical Sector Number of the first 2 KByte block in the DIRBLK band.

.lsnLastDirBlk

This is the Logical Sector Number of the last 2 KByte block in the DIRBLK band.

.lsnDirBlkMap

This is the first sector of the DIRBLK bitmap.

.lsnSidTab

This is the Logical Sector Number of a run of 8 sectors which are used to hold Access Control Information (ACL). In normal HPFS these sectors are reserved and unused. They are only used in HPFS386.

Edit Spare Block

The edit Spare Block function allows you to edit the Spare Block of HPFS disks in their native format. The fields which define the SPAREB structure are defined as follows:

.sig1

This is the first signature value for the Spare Block. It should be F9911849.

.sig2

This is the second signature value for the Spare Block. It should be FA5229C5.

.bFlag

This is the partition status flag. It is bit mapped as follows:

Status Values:

00	Partition is Clean.
01	Partition is Dirty.
02	Spare DIRBLK's are used.
04	Hotfix sectors are used.
08	Bad sector, corrupt disk.
10	Bad bitmap block.
20	Partition was quick formatted.
40	Reserved.
80	An old version wrote to the partition.

.lsnHotFix

This is the Logical Sector Number of the hot fix mapping table.

.culHotFixes

This is the number of entries in the hot fix table which are in use.

.culMaxHotFixes

This is the maximum number of Logical Sectors in the hot fix mapping table. (ie. the maximum number of sectors which can be hot fixed).

.cdbSpares

This is the number of spare DIRBLKs which are free.

.cdbMaxSpare

This is the maximum number of spare DIRBLKs. It is the number of entries in the `alsnSpareDirBlks[]` array which are valid.

.lsnCPInfo

This is the Logical Sector Number of the first code page information sector.

.culCP

This is the total number of code pages used on this volume.

.aulExtra[0]

Under normal HPFS this is unused. Under HPFS386 and LAN Server Version 4, this is the 32 bit CRC of the Super Block. If it does not match the value calculated by CHKDSK then CHKDSK /F is run at boot up.

.aulExtra[1]

Under normal HPFS this is unused. Under HPFS386 and LAN Server Version 4, this is the 32 bit CRC of the Spare Block. If it does not match the value calculated by CHKDSK then CHKDSK /F is run at boot up.

.aulExtra[2-16]

These values are unused and reserved for future expansion.

.sig

This is the signature of the DIRBLK. It should be 77E40AAE.

.offulFirstFree

This is the offset within this DIRBLK to the first free byte.

.culChange

The low bit indicates if this is the topmost DIRBLK in the BTree.

.lsnParent

This is the Logical Sector Number of the parent DIRBLK or, if this is the topmost DIRBLK, then it is the sector number of this directory's FNODE.

.lsnThisDir

This is the Logical Sector Number of this DIRBLK.

The DIRENT fields are as follows:

.cchThisEntry

This is the length of this DIRENT entry. It includes any free space which is on the end of the entry.

The size of a DIRENT is always a multiple of four to force double word alignment. The name is variable length and the DIRENT.BTree down pointer, if present, is the last double word in the entry. Thus there may be padding bytes after the name and before the BTree down pointer. Access Control List information may be stored after the name but before the DIRENT.BTree field. Thus, in order to access the BTree pointer, HPFS must locate it from the end of the DIRENT, rather than from the beginning, based on the name length.

.fFlags

These are the flags for this DIRENT.

.fFlags Values:

01	Special ".." entry.
02	Entry has an ACL.
04	Entry has BTree down pointer (ie. a directory).
08	Entry is a dummy end record
10	Entry has an EA list.
20	Entry has extended perm list.
40	Entry has an explicit ACL.
80	Entry has "needed" EAs

.fAttr

These are the file attributes for this DIRENT.

.fAttr Values:

01	Read Only.
02	Hidden.
04	System.
08	Not used. Would be Volume attribute.
10	Directory.
20	Archive.
40	Long name.
80	Reserved.

.lsnFNode

This is the Logical Sector Number for the FNODE of this entry.

.timLastMod

This is the time and date at which the file was last modified.

.cchFSize

This is the file size.

.timLastAccess

This is the time and date on which the file was last accessed.

.timCreate

This is the time and date on which the file was created.

.ulEALen

This is the number of bytes of Extended Attributes associated with this entry.

.fFlex

This is the "flex" area for this DIRENT. The low three bits are the number of ACL's present. The upper bits are reserved.

.bCodePage

This is the code page index on the volume. It is 7 bits wide. The 8th bit indicates that there are DBC'S present.

.cchName

This is the length of the file name.

.bName

This is the name of the file.

Note:

EXTREME CAUTION should be used when editing this entry. HPFS sorts the DIRENT entries within each DIRBLK based on the file name. Changing the file name may have disastrous effects.

.bTree

This is a BTree pointer to a descendent DIRBLK.

Edit FNODE

The edit FNODE mode allows you to edit FNodes in their native format. The fields which define a FNODE are as follows:

.sig

This is the signature for the FNODE. It should be F7E40AAE.

.ulSRHist

This is the Sequential Read History. It is currently unused.

.ulFRHist

This is the Fast Read History. It is currently unused.

.achName[0] (name length)

This is the length of the file name.

.achName[1] (name itself)

This is the file name itself.

.lsnContDir

This is the Logical Sector Number of the directory FNODE which contains this file or directory. (ie. the parent FNODE).

.aiACL.sp.cbRun

This is the number of sectors used to describe an external ACL definition. The ACL is too large to fit into the FNODE.

.aiACL.sp.lsn

This is the Logical Sector Number of the external (to the FNODE) ACL sector.

.aiACL.usFNL

This is the length of the FNODE resident ACL information.

.aiACL.bDat

This is non zero if the FNODE.aiACL.sp.lsn is an ALSEC. If it is not, then it is direct data.

.cHistBits

This is the count of history bits. It is currently unused.

.aiEA.sp.cbRun

This is the number of sectors used to describe an external EA data definition. The EA data is too large to fit into the FNODE.

.aiEA.sp.lsn

This is the Logical Sector Number of the external (to the FNODE) EA data sector.

.aiEA.usFNL

This is the length of the FNODE resident EA information.

.aiEA.bDat

This is non zero if the FNODE.aiEA.sp.lsn is an ALSEC. If it is not, then it is direct data.

.bFlag

If the low bit is on, this indicates that this is a directory FNODE. Otherwise this is a file FNODE.

.fst.alb.bFlag

This is the flag byte to describe the B+ tree information.

0x20 Parent is an FNODE
0x80 Following are ALNODE structures.

If the ALNODE bit is not set, then the following are ALLEAF structures.

.fst.alb.bPad

This is used to align to a 32 bit boundary.

.fst.alb.cFree

This is a count of the number of free entries left in the array.

.fst.alb.cUsed

This is a count of the number of entries in the array which are in use.

.fst.alb.oFree

This is the offset to the next free element in the array.

.fst.a.aall[0-7].lsnLog LEAF FNODE

This is the logical sector offset within the file. This number starts at zero.

.fst.a.aall[0-7].csecRun LEAF FNODE

This is the length, in sectors, of the block of data on the disk.

.fst.a.aall[0-7].lsnPhys LEAF FNODE

This is the sector number on the disk of the first sector of the block. If the FNODE corresponds to a directory, then this field is the LSN of the topmost DIRBLK in the BTree.

.fst.a.aaln[0-11].lsnLog NODE FNODE

This is the logical sector offset within the file.

.fst.a.aaln[0-11].lsnPhys NODE FNODE

This is a Logical Sector Number of a sector on disk which contains an array of ALNODES or an array of ALLEAFs (ie. an ALSEC).

.fst.ulVLen

This is the length of valid data within the file.

.ulRefCount

This is the number of "Needed" EA's for this file.

.achUID

This is the User ID. It is currently unused.

.usACLBase

This is the offset into the FNODE of the first ACL entry (if present) or EA.

.abSpare

This is currently unused.

Edit ALSEC

The edit ALSEC mode enables you to edit an ALSEC sector in its native format. The `.alb.bFlag` variable defines whether an ALSEC contain either 40 ALLEAF structures or 60 ALNODE structures. The fields which define an ALSEC are as follows:

.sig

This is the signature for the ALSEC. It should be 37E40AAE.

.lsnSelf

This is the sector of the allocation block itself. This is used for allocating other blocks near the current one.

.lsnRent

This is the Logical Sector Number of the parent of this sector. The ALSEC.`alb.bFlag` byte indicates if the parent is either a FNODE or another ALSEC.

.alb.bFlag

This is the flag byte to describe the B+ tree information.

0x20	Parent is an FNODE
0x80	Following is ALNODE structures.

If the ALNODE bit is not set, then the following are ALLEAF structures.

.alb.bPad

This is used to align to a 32 bit boundary.

.alb.cFree

This is a count of the number of free entries left in the array.

.alb.cUsed

This is a count of the number of entries in the array which are in use.

.alb.oFree

This is the offset to the next free element in the array.

.a.aall[0-39].lsnLog LEAF ALSEC

This is the logical sector offset within the file. This number starts at zero.

.a.aall[0-39].csecRun LEAF ALSEC

This is the length, in sectors, of the block of data on the disk.

.a.aall[0-39].lsnPhys LEAF ALSEC

This is the sector number on the disk of the first sector of the block. If the FNODE corresponds to a directory, then this field is the LSN of the topmost DIRBLK in the BTree.

.a.aaln[0-59].lsnLog NODE ALSEC

This is the logical sector offset within the file.

.a.aaln[0-59].lsnPhys NODE ALSEC

This is a Logical Sector Number of a sector on disk which contains an array of ALNODES or an array of ALLEAFs (ie. an ALSEC).

.bPad

Twelve bytes of padding to pad it out to a sector. This is unused.

Edit Code Page Information

The edit Code Page Information mode allows you to edit code page information sectors in their native format. Each code page information sector has a small header followed by up to 31 CPINFOENTRY structures. The fields which define a CODEPAGEINFOSECTOR are as follows:

.sig

This is the signature of the Code Page Information Sector. It should be 494521F7.

.cCodePage

This is the count of code page information entries which follow in this sector.

.iFirstCP

The volume specific index of the first code page in the array of code page information entries which follow. For this first code page info sector this will be zero, for the next one this will be 31, and so on.

.lsnNext

This is the sector number of the next code page information sector. If there are no additional code page information sectors, then this will be zero.

Each CPINFOENTRY structure is defined as follows:

.CPInfoEnt[0-30].usCountryCode

This is the country code of this character set.

.CPInfoEnt[0-30].usCodePageID

This is the code page for this character set.

.CPInfoEnt[0-30].cksCP

This is the checksum of corresponding code page data entry. The code page data entry appear in a code page data sector pointed by CPINFOENTRY.lsnCPData.

.CPInfoEnt[0-30].lsnCPData

This is the Logical Sector Number of code page data sector which holds the code page data entry for this character set.

Note:

A Code Page Data Sector can hold three separate code pages.

.CPInfoEnt[0-30].iCPVol

This is the volume specific index of this character set.

.CPInfoEnt[0-30].cDBCSRange

This is the count of DBCS ranges in this character set.

Edit Code Page

The edit Code Page mode allows you to edit code page data sectors in their native format. Each code page data sector has a small header followed by up to three CPDATAENTRY structures. The fields which define a CPDATASECTOR are as follows:

.sig

This is the signature for the Code Page Data Sector. It should be 894521F7.

.cCodePage

This is the count of code page data entries in this sector.

.iFirstCP

This is the volume specific character set index of first code page data entry in this sector.

.cksCP[0]

This is the checksum of the first code page data entry contained in this sector.

.cksCP[1]

This is the checksum of the second code page data entry contained in this sector.

.cksCP[2]

This is the checksum of the third code page data entry contained in this sector.

.offCPData[0]

This is the offset of the first code page data entry.

.offCPData[1]

This is the offset of the second code page data entry.

.offCPData[2]

This is the offset of the third code page data entry.

Each CPDATAENTRY is defined as follows:

.CPData[0-2].usCountryCode

This is the country code for this character set.

.CPData[0-2].usCodePageID

This is the code page for this character set.

.CPData[0-2].cDBCSRange

This is the count of the DBCS ranges. This will specify the number of DBCS range pairs in the DBCSRange array.

.CPData[0-2].bCaseMapTable (128-143)

This is the 16 byte mapping table for characters 128 to 143 in this character set.

.CPData[0-2].bCaseMapTable (144-159)

This is the 16 byte mapping table for characters 144 to 159 in this character set.

.CPData[0-2].bCaseMapTable (160-175)

This is the 16 byte mapping table for characters 160 to 175 in this character set.

.CPData[0-2].bCaseMapTable (176-191)

This is the 16 byte mapping table for characters 176 to 191 in this character set.

.CPData[0-2].bCaseMapTable (192-207)

This is the 16 byte mapping table for characters 192 to 207 in this character set.

.CPData[0-2].bCaseMapTable (208-223)

This is the 16 byte mapping table for characters 208 to 223 in this character set.

.CPData[0-2].bCaseMapTable (224-239)

This is the 16 byte mapping table for characters 224 to 239 in this character set.

.CPData[0-2].bCaseMapTable (240-255)

This is the 16 byte mapping table for characters 240 to 255 in this character set.

.CPData[0-2].DBCSRange[0].ucStart

This is the start byte of the DBCS lead byte range.

.CPData[0-2].DBCRange[0].ucEnd

This is the end byte of the DBCS lead byte range.

Edit Bad Block List

The edit Bad Block List mode enables you to edit the bad block list in its native format. The BADBLOCKLIST structure has the following fields:

.lsnNextSector

This is the Logical Sector Number of the next BADBLOCKLIST structure. If it is zero, then there are no more entries.

.lsnBadSector[0-510]

The Logical Sector Number of sectors which have been marked bad.

Edit Hot Fix List

The edit Hot Fix List mode allows you to edit the dynamic hot fix list in its native format. The length of each array is defined by the SPAREB.culMaxHotFixes item. The HOTFIXLIST structure is defined as follows:

.lsnFrom[]

This is the Logical Sector Number of a sector which has been marked bad dynamically.

.lsnTo[]

This is the Logical Sector Number of a sector which has been reserved to act as a spare when a dynamic hot fix occurs.

.lsnFNode[]

This is the Logical Sector Number of the FNODE of the file or directory that owns the damaged sector.

Edit Bitmap List

The edit Bitmap List mode enables you to edit the bitmap lists in their native format. Each BITMAPLIST structure is four sectors long. If the list needs to be greater than four sectors long, it will flow onto the next four contiguous sectors. The list is terminated by a zero entry.

.lsnStartBitmapSector[0-511]

This is the Logical Sector Number of first sector of a bitmap band.

Appendix J - Internet Services

Summary



WarpSpeed Computers
Internet Support - Australia



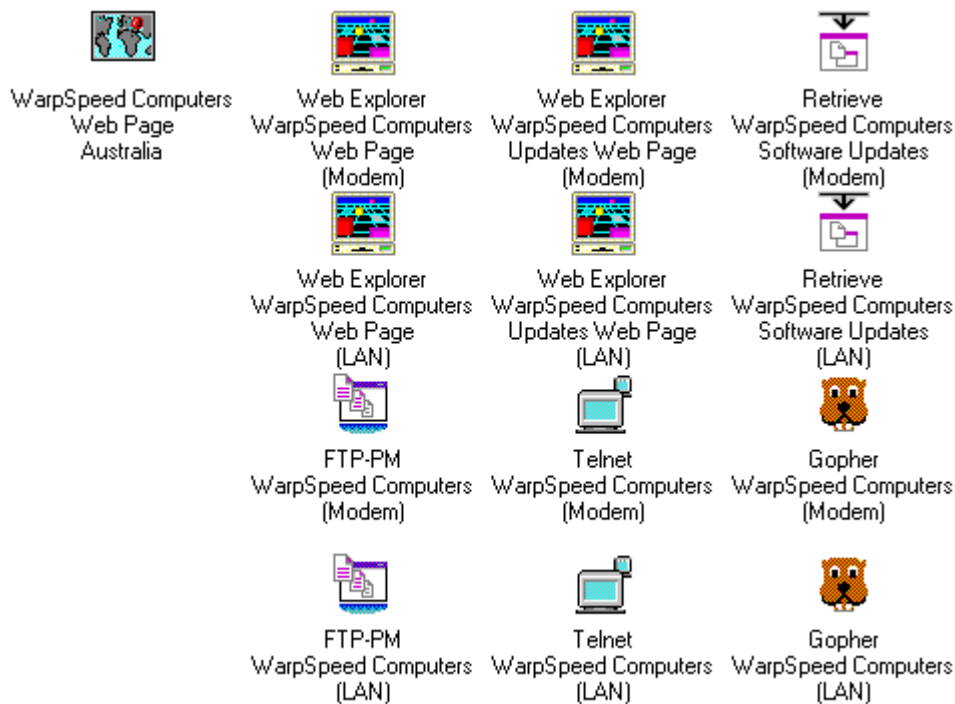
WarpSpeed Computers
Internet Support - USA

WarpSpeed Computers is committed to providing the very best support available. A major portion of this is provided electronically. To this end, WarpSpeed Computers has set up multiple World Wide Web and other servers around the globe to ensure ease of accessibility. With servers based in Australia, Europe and North America, there should be a server near you. For the fastest connection, you should choose the one closest to you.

IBM has chosen not to document the interface to FTP and TELNET objects. So the details are described below. You may wish to manually create a FTP and/or a TELNET object using templates.

Available Services

The following services are available via the Internet.



The services listed as (LAN) are for users who have Internet access via a LAN. The services listed as (Modem) are for users who use direct dial-up access to the Internet. The default dialer is SLIPPM (as used by LINKUP.EXE). You may wish to change this.

WWW



This entire manual, as well as contact information, reseller sites, updates, hints and tips are available from the web sites. The addresses are as listed below:

Asia/Pacific
North America

<http://www.warpspeed.com.au>
<http://www.netins.net/showcase/spectre>

FTP



The following FTP sites are also available:

Asia/Pacific

<ftp://ftp.warpspeed.com.au>

North America

<ftp://ftp.netins.net/showcase/spectre/.gopher>

GOPHER



Gopher is only available from Australia. Its address is:

Asia/Pacific

<gopher://gopher.warpspeed.com.au>

TELNET



A TELNET server has been provided to provide for those of you who like the BBS style of interface.

Asia/Pacific

<telnet://bbs.warpspeed.com.au>

UPDATES



An updates server has been provided. This will be the first place where new CSD's will be available. It is recommended that you check on a regular basis, (say monthly), to see if any new updates have been produced. An announcement will also be made in the COMP.OS.OS2.ANNOUNCE news group when updates are made available.

Asia/Pacific

<gopher://updates.warpspeed.com.au:77>

Manually installing a CSD file

The preferred method of installing CSD's is to use the Retrieve Software Updates icon in the Internet Support folder of your region. However, if you can not do this (Merlin may not, as of this time of writing, include UPDATE.EXE) or you have obtained the CSD file by another means, then please follow these instructions. The files can also be obtained using a standard GOPHER program connecting to the UPDATES address above - using port 77 instead of the normal port 70.

Step 1.

Copy the GUffyddd.ZIP file into the directory in which you installed the utilities.

Step 2.

UnZip the file. There should be two files: GUffyddd.CSD and UPDINST.CMD.

Step 3.

Enter: UPDINST.CMD at a command prompt.

The installation of the CSD should proceed as described in the CSDInst chapter.

Other sites

Updates may also be available from the following sites:

Site	Country	Location or URL
CompuServe	Global	Library 4 in the OS2USER forum
Hobbes	USA	ftp://ftp-os2.nmsu.edu
Walnut Creek	USA	ftp://ftp-os2.cdrom.com
University of Stuttgart	Germany	ftp://ftp.uni-stuttgart.de
Latrobe University	Australia	ftp://ftp.latrobe.edu.au

Glossary

access mode	When a process creates or opens a file, it specifies the access and share modes of the file. The access mode specifies whether the file is for reading, writing or both.
and	<p>A logical operation between two numbers.</p> $0 \text{ AND } 0 = 0$ $0 \text{ AND } 1 = 0$ $1 \text{ AND } 0 = 0$ $1 \text{ AND } 1 = 0$ <p>See also OR, XOR.</p>
archive	An archive is a special file which is managed by an archive program. A set of files is merged into a single file. As each file is added to the archive it may be compressed so that the size of the archive is smaller than the sum of all of its components. There are various methods of compression. Each having their own strengths and weaknesses. Some of the common archive formats are: ARC, ARJ, PAK, LZH, ZIP and ZOO.
archiver	A program which manages archive files.
batch file	A batch file is a text file which contains a sequence of commands to be executed. Batch files do not necessarily execute in sequential order. They may loop or branch either unconditionally or in response to the exit code from a program.
binary	A number system which uses a base of 2. The valid numbers are 0 or 1.

bit	BIT stands for Binary digIT. A bit is the smallest amount of information which a computer can process. It may have a value of either 0 or 1. Computers store information in groups of bits, these groups may be nibbles, bytes, words or double words.
BBS	Bulletin Board System. A BBS is a computer program which enables people with a MODEM to connect two computers and transfer information between them. This information may include files, electronic mail or both.
boot sector	A boot sector is the first sector (logical sector one) on a disk volume. It contains information about the disk and a small bootstrap program.
bootstrap	A bootstrap is a small program which is contained in your boot sector. This program may "boot" your computer by loading the operating system, or it may simply display the message which you see when you attempt to boot off a non system disk.
byte	A byte is a group of 8 bits. A signed byte contains values from -127 to 128.
cache	The use of memory to buffer data which has to be transferred to and from a device which is slower than system memory.
CBCS	Single Byte Character Set. This character encoding scheme uses 8 bits to define a single character. This provides us with 256 different characters.
checksum	A checksum is a number used to check the integrity of a file. There are numerous methods for calculating a checksum value.

cluster	A cluster is the minimum number of sectors which are allocated on a logical disk volume. A file consists of one or more clusters. The number of sectors in a cluster is always a power of two.
cmd file	An OS/2 batch file has a .CMD extension.
command interpreter	The command interpreter is a program which reads the keyboard and executes the commands which the user enters. It is only used in either windowed or full screen command prompt sessions. Typically the command interpreter is CMD.EXE
compression	The method of shrinking a series of bytes to smaller than the total number of bytes.
CPU	Central Processing Unit. The piece of hardware which controls the computer. It executes the programs which run on a computer. OS/2 runs on the Intel 80286, 80386, 80486 and Pentium series of microprocessor CPU's.
DBCS	Double Byte Character Set. This character encoding scheme uses 16 bits to define a single character. This provides us with 65,536 different characters. It is used for all of the Asian languages as well as the Arabic and Hebrew ones.
decimal	A number system which uses a base of 10. The valid numbers are 0 to 9.
directory	A named group of files in a file system.
directory tree	An outline of all directories in a file system.

DLL	A Dynamic Link Library is a special file which contains subroutines that are bound to an application program at runtime, not link time. This may save disk space by placing commonly used routines in a DLL, rather than placing copies of all of the routines in each individual program.
double word	A group of thirty two (32) bits. When used as a signed variable it has a range of values from -2,147,483,648 to 2,147,483,647 and 0 to 4,294,967,295 as an unsigned number.
dynamic hot fix	Some file systems, such as HPFS, support dynamic hot fixes. When a sector on a disk has a correctable error the data will be moved to a good sector on the disk and the problem area will be marked as bad. This process of transferring data from bad to good portions of the disk whilst the disk is in use, is known as dynamic hot fixing.
environment variables	Environment variables are a group of strings which are associated with each process. You can view the entire group of environment variables by entering the SET command at any command prompt.

errorlevel	<p>This is a special statement in batch files which enable conditional jumping on the exit code returned from a program. They are used in conjunction with an IF statement in a batch file. The errorlevel statement acts as a "greater than" or "equal to" for the specified error level. For example:</p> <pre data-bbox="703 869 1326 1189"> BE DAY if errorlevel 3 goto Finish if errorlevel 2 goto Second :First echo Today is the first day of the month goto Finish :Second echo Today is the second day of the month :Finish </pre>
exit code	<p>When a program terminates, it returns an exit code which may be registered by the command interpreter as an errorlevel.</p>
extended attributes	<p>See appendix A for a complete description of extended attributes.</p>
FAT	<p>File Allocation Tables. A method which uses tables to define the clusters used by files and directories.</p>
fidonet	<p>A BBS network which started in the US in the mid eighties. It is run by individuals who fund the cost of running the BBS themselves. Currently, there are over twenty thousand fidonet BBS's world wide.</p>

file attributes	<p>Attributes attached to a file. The attributes may include:</p> <p>Read Only Indicating that the file may only be opened for reading.</p> <p>Archive Indicating that the contents of the file have changed and that it needs to be backed up.</p> <p>System Indicates that the file is part of the operating system.</p> <p>Hidden Indicates that the file should not be listed in normal directory listings.</p>
file system	<p>The part of the operating system which manages directories and files. It can translate file access calls into requests which the disk device drivers can use to transfer data to and from logical disks.</p>
general protection fault	<p>An error which occurs when a program executes an invalid instruction or uses an invalid memory address. OS/2 takes control and terminates the application when a GP fault occurs. It is also known as a TRAP D, referring to the interrupt number for the Intel series of processors which is called when a GP fault occurs.</p>
hexadecimal	<p>A number system which uses a base of 16. The valid digits are 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E and F, representing the numbers zero to fifteen. Each hexadecimal number is represented by a group of four bits. It is used heavily in the computer world as it means an eight bit byte can be represented by two hexadecimal digits.</p>

HPFS	High Performance File System. A special file system for OS/2 which uses caching and different allocation strategies to achieve significant performance increases over the FAT file system. HPFS also allows file names of up to 254 characters.
integer	A group of sixteen bits. It is a signed number and has a range of values from -32,768 to 32,767.
logical disk	Is represented by a letter and a colon. For example A:. It contains a boot sector, root directory and zero to many subdirectories. Multiple logical drives may span a single physical drive.
LSN	Logical Sector Number. This is the sector number of a sector from the beginning of a logical disk, as opposed to a physical disk.
Maximus BBS	A BBS system written by Scott Dudley. It is available for both the OS/2 and DOS operating systems.
MIME	Multipurpose Internet Mail Extensions. A format for imbedding binary and others types of files in EMail messages.
multitasking	The process of sharing a processor among multiple processes so that they all appear to be running simultaneously.
MSB	Most Significant Bit, or the right most bit. It may be used as a sign bit to allow signed numbers.
nibble	A group of 4 bits. A nibble represents a single hexadecimal digit. As a number it ranges from 0 to 15 in decimal or 0 to F in hexadecimal.

number systems

Number systems are means which we use to represent numbers. People use the decimal number system, but computers use binary number system. The octal and hexadecimal number systems are directly related to the binary number system. They are all a power of two. The way in which they are used is still the same. For example the number 123 decimal is represented as:

$$\begin{array}{r} 10^2 \quad 10^1 \quad 10^0 \\ 100 \quad 10 \quad 1 \\ 1*100 \quad 2*10 \quad 3*1 \\ 1 \quad 2 \quad 3 \end{array}$$

or 123 decimal.

The same number in binary would be expressed as:

$$\begin{array}{r} 2^7 + 2^6 + 2^5 + 2^4 + 2^3 + 2^2 + 2^1 + 2^0 \\ 128 \quad 64 \quad 32 \quad 16 \quad 8 \quad 4 \quad 2 \quad 1 \\ 0*128 \quad 1*64 \quad 1*32 \quad 1*16 \quad 1*8 \quad 0*4 \quad 1*2 \quad 1*1 \\ 0 \quad 1 \quad 1 \quad 1 \quad 1 \quad 0 \quad 1 \quad 1 \end{array}$$

or 01111011 binary.

The same number in hexadecimal would be expressed as:

$$\begin{array}{r} 16^1 + 16^0 \\ 16 \quad 1 \\ 7*16 \quad B*1 \\ 7 \quad B \end{array}$$

or 7B hexadecimal.

octal

A number system which uses a base of 8. The valid digits are 0, 1, 2, 3, 4, 5, 6 and 7. Each octal number is represented by a group of three bits.

or	<p>A logical operation between two numbers.</p> <p>0 OR 0 = 0 0 OR 1 = 1 1 OR 0 = 1 1 OR 1 = 1</p> <p>See also AND, XOR.</p>
partition	<p>A physical fixed size allocation of disk space. Logical disks reside in partitions.</p>
physical disk	<p>A physical or real disk attached to your system. A single physical disk may be partitioned into multiple logical disks.</p>
preemptive multitasking	<p>The ability of the CPU to share itself among multiple processes without their cooperation.</p>
Presentation Manager	<p>The graphical user interface for OS/2.</p>
priority	<p>A number associated with each thread or process. Processes with a higher priority receive more CPU than threads with a lower priority.</p>
process	<p>An instance of an executing program. A process may own, files, pipes, semaphores, memory as well as other system resources.</p>
root directory	<p>The first directory on a disk within which all files and other subdirectories exist.</p>
sector	<p>The smallest area on a disk which can be either read or written by a disk drive. It is usually 512 bytes, but it may be 128, 256, 1024 or 2048.</p>
semaphore	<p>An interprocess communication mechanism which has only two states. It is typically used to signal ownership of a resource.</p>
session	<p>See process.</p>

share mode	When a process creates or opens a file, it specifies the access and share modes of the file. The share mode defines how other processes can access the file. Other processes may have, read and write, read only, write only or no access to the file.
signed number	A signed number is a number where the MSB is used as the sign bit. This allows the number to hold negative numbers.
slack	The difference between the actual file size and the amount of space allocated to the file by the file system.
stack fault	A CPU exception which occurs when a program uses all of its assigned stack space. It is also known as a TRAP E.
subdirectory	A group of files or directories which are located under the root directory.
system semaphore	A system semaphore is a special form of semaphore that has the form: <code>\SEM\PATH\FILENAME.EXT</code> Where the PATH and .EXT portions are optional.
task	See process.
thread	The point of execution within a process. A process has at least one thread.
timer tick	A hardware interrupt which occurs periodically to enable OS/2 to monitor the system.
timeslice	The amount of time that OS/2 allocates to a thread before having to preempt it.
time slicing	See preemptive multitasking.

UNIX	An operating system originally designed by AT&T.
unsigned number	An unsigned number is a number which has no sign bit. Its value ranges from zero to the maximum value allowed by the size of the number.
volume label	A special file name which may be used to label a logical disk. It has a maximum of eleven characters.
write-through	A cache mode. Data is not held in cache memory, it is transferred directly to the device.
word	A group of sixteen bits. It is an unsigned number and has a range of values from 0 to 65,535.
xor	Exclusive OR. A logical operation between two numbers. $0 \text{ XOR } 0 = 0$ $0 \text{ XOR } 1 = 1$ $1 \text{ XOR } 0 = 1$ $1 \text{ XOR } 1 = 0$ See also AND, OR.

License Agreement

SOFTWARE LICENSE AGREEMENT OF WARPSPEED COMPUTERS

WARPSPEED COMPUTERS ("Licensor") IS WILLING TO LICENSE THE ENCLOSED SOFTWARE TO YOU ONLY IF YOU ACCEPT ALL OF THE TERMS IN THIS LICENSE AGREEMENT. PLEASE READ THE TERMS CAREFULLY BEFORE YOU OPEN THIS PACKAGE, BECAUSE BY OPENING THIS SEALED DISK PACKAGE YOU ARE AGREEING TO BE BOUND BY THE TERMS OF THIS AGREEMENT. IF YOU DO NOT AGREE TO THESE TERMS, LICENSOR WILL NOT LICENSE THIS SOFTWARE TO YOU, AND IN THAT CASE YOU SHOULD RETURN THIS PRODUCT PROMPTLY, INCLUDING THE PACKAGING, THIS UNOPENED DISK PACKAGE, AND ALL WRITTEN MATERIALS, TO THE PLACE OF PURCHASE PROMPTLY FOR A FULL REFUND.

Ownership of the Software

1. The enclosed Licensor software program ("Software") and the accompanying written materials are owned by Licensor and are protected by Australian copyright laws, by laws of other nations, and by international treaties.

Grant Of License

2. Licensor grants to you the right to use one copy of the Software on a single computer. You may load one copy into permanent memory of one computer and may use that copy, or the enclosed diskettes, only on that same computer. You may install the Software on a single network server, provided that you have a License for Network Station from Licensor for each station of the network at which the Software is used.

3. This license is valid worldwide.

Restrictions on Use and Transfer

4. This Software package contains either (i) one 3.5" disk and one CD-ROM or, (ii) 3.5" disks only, you may use only the size disks appropriate for your computer. You may not use the other size disks on another computer or loan, rent, transfer, or assign them to another user except as part of the permanent transfer of the Software and all written materials (as provided for below).

5. You may not copy the Software, except that (1) you may make one copy of the Software solely for backup or archival purposes, and (2) you may transfer the Software to a single hard disk provided you keep the original solely for backup or archival purposes. You may not copy the written materials.

6. You may permanently transfer the Software and accompanying written materials (including the most recent update and all prior versions) if you retain no copies and the transferee agrees to be bound by the terms of this Agreement. Such a transfer terminates your license. You may not rent or lease the Software or otherwise transfer or assign the right to use the Software, except as stated in this paragraph.

7. You may not reverse engineer, decompile, or disassemble the Software.

Limited Warranty

8. Licensor warrants that the Software will perform substantially in accordance with the accompanying written materials for a period of 90 days from the date of your receipt of the Software. Any implied warranties on the Software are limited to 90 days. Some states do not allow limitations on duration of an implied warranty, so the above limitation may not apply to you.

9. LICENSOR DISCLAIMS ALL OTHER WARRANTIES, EITHER EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, AND NON-INFRINGEMENT, WITH RESPECT TO THE SOFTWARE AND THE ACCOMPANYING WRITTEN MATERIALS. This limited warranty gives you specific legal rights. You may have others, which vary from state to state.

10. LICENSOR'S ENTIRE LIABILITY AND YOUR EXCLUSIVE REMEDY SHALL BE, AT LICENSOR'S CHOICE, EITHER (A) RETURN OF THE PRICE PAID OR (B) REPLACEMENT OF THE SOFTWARE THAT DOES NOT MEET LICENSOR'S LIMITED WARRANTY AND WHICH IS RETURNED TO LICENSOR WITH A COPY OF YOUR RECEIPT. Any replacement Software will be warranted for the remainder of the original warranty period or 30 days, whichever is longer. These remedies are not available outside Australia.

11. This Limited Warranty is void if failure of the Software has resulted from modification, accident, abuse, or misapplication.

12. IN NO EVENT WILL LICENSOR BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY LOSS OF PROFITS, LOST SAVINGS, OR OTHER INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF YOUR USE OR INABILITY TO USE THE SOFTWARE. Because some states do not allow the exclusion or limitation of liability for consequential or incidental damages, the above limitation may not apply to you.

13. This Agreement is governed by the laws of the State of Victoria, Australia.

14. If you have any questions concerning this Agreement or wish to contact Licensor for any reason, please write: WarpSpeed Computers, PO Box 212, Brunswick VIC 3056, Australia or call +61-3-9384-1060.

15. U.S. Government Restricted Rights. The Software and documentation are provided with Restricted Rights. Use, duplication, or disclosure by the Government is subject to restrictions set forth in subparagraph (c)(1) of The Rights in Technical Data and Computer Software clause at DFARS 252.227-7013 or subparagraphs (c)(1)(ii) and (2) of Commercial Computer Software - Restricted Rights at 48 CFR 52.227-19, as applicable. Supplier is WarpSpeed Computers, Unit 4, 1 Mitchell St, Brunswick, VIC 3056, Australia.