

Zap

User's Reference Manual

188-190-205 A

May 1985

WICATsystems



Copyright © 1985 by WICAT Systems Incorporated
All Rights Reserved
Printed in the United States of America

Receipt of this manual must not be construed as any kind of commitment,
on the part of WICAT Systems Incorporated, regarding delivery or
ownership of items manufactured by WICAT.

This manual is subject to change without notice.

first printing May 1985

Typographical Conventions Used in this Publication

Bold facing indicates what you should type.

Square brackets, [], indicate a function key, the name of which appears in uppercase within the brackets. For example, [RETRN], [CTRL], etc.

Underlining is used for emphasis.

Information about this Manual

Review the following items before you read this publication.

The subject of this manual

This manual explains what the Zap program is, how to execute the Zap program, and how to execute each function within the Zap program.

The audience for whom this publication was written

This manual is for the system manager or systems programmer who needs to edit files that contain binary data. Before you read this manual, you should complete the tutorials in either the WMCS User's Introductory Manual or the WMCS System Manager's Introductory Manual.

Related publications

This manual describes a system utility. The Reader's Guide to WMCS Publications shows you the relation of this manual to other WMCS manuals.

Table of Contents

Chapter 1 General Information

The Zap Program and Your Terminal Type	1-2
How to Execute the Zap Program from the CIP	1-2
The Zap Program and Your Terminal Screen	1-6
The Text Field.....	1-7
The Hexadecimal Field.....	1-8
Byte Addresses.....	1-9
Page Numbers.....	1-10
The Filename.....	1-11
Editing a Disk Device.....	1-13
How to Execute Zap Functions	1-14
The Zap Function Line.....	1-14
Single-keystroke Commands.....	1-15
Moving the Cursor	1-16
Making Changes	1-16
Help Display	1-17
Unknown Functions	1-17
The Minus Sign	1-17
Restoring Deleted Material	1-17
Repeating and Editing the Function Line	1-18
File Versions	1-18
Diagnostic Messages	1-18

Chapter 2 Directory of Zap Functions

Move the Cursor to:	2-2
Compute a checksum for:	2-2
Copy	2-2
Search	2-3
Create a CIP	2-3
Undo Changes	2-3
Create a Print-file of:	2-3
Set to a Constant Value	2-3
Screen Functions and Help Display	2-4

Table of Contents

File Control.....	2-4
Toggle Functions.....	2-4
Repeat or Edit the Previous Escape-key Function.....	2-4
Control-characters on the Function Line.....	2-4
Terminate a Zap Function.....	2-5

Chapter 3 Dictionary of Zap Functions

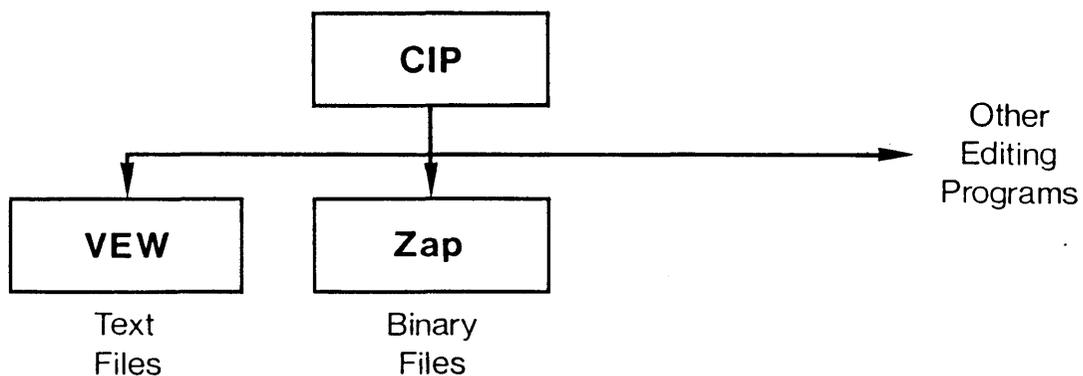
ab	Abort Zap
bc	Compute boot block checksum
co	Copy page
cp	Create a CIP
cs	Copy sector
ef	Edit new file or device
el	Move to end of line
ex	Exit Zap and save changes
fc	Compute file control block checksum
fl	Move to front of line
go	Move to specific sector or page
he	Display help file
mc	Move to next or previous character
me	Move to end of file
ml	Move to front of next or previous line
mp	Move to next or previous page
ms	Move to next or previous sector
mt	Move to top of file
mw	Move to next or previous word
oc	Compute other boot block checksum
pl	Move to same position in next or previous line
pr	Create print-file
rp	Repeat last command
rs	Refresh screen
sa	Save changes
sp	Set page to constant
sr	Search
ss	Set sector to constant
tm	Turn mask8bit display flag on or off
ud	Undo changes

Chapter 1
General Information

The Zap program, one of the utilities in the WMCS operating system, displays the contents of a file or device in hexadecimal format and allows you to edit any file on a disk or to examine but not change a file on a tape.

Zap is particularly useful to the system manager or programmer who needs to edit files that contain binary data, such as executable files (i.e., files with a .EXE extension), directory files (i.e., files with a .DIR extension), and certain system files. For example, Zap is the only program by which you can edit the files /ROOTDIR/FCB.SYS, /ROOTDIR/FCBBITMAP.SYS, and /ROOTDIR/BITMAP.SYS. Zap can also give you direct access to the physical sectors on a disk.

The following illustration shows the relationship of the Zap program to other programs on your system.



The Zap program (like any other WMCS utility) is executed from the Command Interpreter Program (CIP). In other words, each time you type the CIP command mnemonic `zap` onto the CIP command line, you gain access to the Zap program.

General Information

The functions explained in this manual apply only after you execute the `zap` command.

NOTE: Even though Zap is a very useful program, unless you know precisely what you are doing you may destroy the means of retrieving a file (or the files within a particular directory and its subdirectories).

The Zap Program and Your Terminal Type

The Zap program works with a variety of terminal types. You must tell Zap what type of terminal you are using so it can adapt the Zap program to your particular type of terminal.

To find out how to set your terminal type, read the description of the `DSTAT` command in the WMCS User's Reference Manual. For further information on terminal types, refer to the VEW User's Reference Manual. Zap uses the same terminal setup files as VEW.

How to Execute the Zap Program from the CIP

A summary of how to execute the `zap` command, useful for quick reference, is provided in the WMCS User's Reference Manual. A complete description of executing the Zap program from the CIP is given here.

Command Line Syntax

Mnemonic	zap		
Required parameter	Filename		
Switches	:mask8bit	:memory=	:readonly
	:rewind	:setupin=	:setupout=

Parameters

Filename	Function	Use this parameter to specify the name of the file or the name of the disk device you want to edit with Zap.
	Default	None.
	Syntax	Type a single, standard filename, or the name of a mounted disk device preceded by an underscore. Wildcard symbols are disallowed. If a tape file is specified, only the :readonly mode is allowed.

Switches

:mask8bit	Function	Use this switch to mask the high order or eighth bit for the text display of the file. The hexadecimal part of the display will not be affected.
	Default	Do not mask the eighth bit.
	Syntax	Type :mask8bit
:memory=	Function	Use this switch to specify the amount of main memory (in kilobytes) to use as a buffer for holding modified pages while editing in Zap.
	Default	32 Kilobytes. A value of 0 means no limit other than available physical or logical memory.
	Syntax	Type :memory= followed by an integer between 0 and 1000.
:readonly	Function	Use this switch to prevent modification to your files or devices in Zap. If :readonly is specified, then when the file appears on your screen, Zap allows you to move the cursor in the file, but it does not allow you to make any changes. A bell sounds if you try to type new characters or to use any modification commands.
	Default	:noreadonly. However, if you specify a tape file, the default is :readonly.
	Syntax	Type :readonly

General Information

<code>:rewind</code>	Function	Use this switch to rewind the tape device.
	Default	<code>:norewind</code> , i.e., the tape device begins at the current position instead of at the beginning of the tape.
	Syntax	Type <code>:rewind</code>
<code>:setupin=</code>	Function	Use this switch to specify an alternate setup file to use for <code>sys\$input</code> . If this name specifies a directory path, it will look in that directory for the setup file based on the terminal type. If this specifies an explicit file, it will use that file.
	Default	Use the setup file in <code>sys\$disk/syslib.setup/</code> determined by the terminal type, e.g., if your terminal is a T7000, then the file used will be <code>sys\$disk/syslib.setup/setup252.sys</code> .
	Syntax	Type <code>:setupin=</code> followed by a filename.
<code>:setupout=</code>	Function	Use this switch to specify an alternate setup file to use for <code>sys\$output</code> . If this name specifies a directory path, it will look in that directory for the setup file based on the terminal type. If this specifies an explicit file, it will use that file.
	Default	Use the setup file in <code>sys\$disk/syslib.setup/</code> determined by the terminal type, e.g., if your terminal is a T7000, then the file used will be <code>sys\$disk/syslib.setup/setup252.sys</code> .
	Syntax	Type <code>:setupout=</code> followed by a filename.

Examples

> `zap command.doc`

This command allows you to examine or modify the most recent version of the file `COMMAND.DOC`.

> `zap command.doc :readonly`

This command is like the first example except it does not allow you to modify the file.

> zap _mt0/rootdir/fcb.dat

This command allows you to examine but not modify the file /ROOTDIR/FCB.DAT on whatever tape is loaded in the device _MT0. Because the file is located on a tape device, the Zap program automatically uses the readonly mode.

> zap _dx0

This command allows you to examine or modify the physical sectors of the device _DX0.

Notes on Usage

If you use Zap to directly access physical sectors on a disk, as in the last example above, you must have the privileges READPHYS and WRITEPHYS. Refer to the WMCS System Manager's Reference Manual for information on process privileges and file protection.

Related CIP commands

dump Display the contents of a file in hexadecimal format

General Information

The Zap Program and Your Terminal Screen

When you type the `zap` command and a filename onto the CIP command line and strike [RETRN], your terminal screen looks like the following example.

```
00000000 73 65 64 2E 0A 54 68 65 20 66 69 72 73 74 20 72 sed..The first r
00000010 65 6C 65 61 73 65 20 6F 66 20 74 68 69 73 20 73 elease of this s
00000020 6F 66 74 77 61 72 65 20 77 69 6C 6C 20 75 73 65 oftware will use
00000030 20 65 74 68 65 72 6E 65 74 20 61 73 20 74 68 65 ethernet as the
00000040 0A 68 61 72 64 77 61 72 65 20 70 72 6F 74 6F 63 .hardware protoc
00000050 6F 6C 2E 0A 0A 49 6E 20 61 64 64 69 74 69 6F 6E cl...In addition
00000060 20 74 6F 20 74 68 65 20 6E 65 74 77 6F 72 6B 69 to the networki
00000070 6E 67 20 66 65 61 74 75 72 65 73 2C 20 62 75 67 ng features, bug
00000080 20 66 69 78 65 73 20 61 6E 64 20 65 6E 68 61 6E fixes and enhan
00000090 63 65 6D 65 6E 74 73 0A 73 69 6E 63 65 20 57 4D cements.since NM
000000A0 43 53 20 35 2E 30 20 61 72 65 20 69 6E 63 6F 72 CS 5.0 are incor
000000B0 70 6F 72 61 74 65 64 2E 20 20 4F 74 68 65 72 20 porated. Other
000000C0 69 6D 70 6F 72 74 61 6E 74 20 65 6E 68 61 6E 63 important enhanc
000000D0 65 6D 65 6E 74 73 20 74 6F 0A 57 4D 43 53 20 69 ements to.NPKS i
000000E0 6E 63 6C 75 64 65 3A 0A 0A 09 31 2E 20 41 20 6E nclude:...l. A n
000000F0 65 77 20 73 63 68 65 64 75 6C 65 72 20 69 73 20 ew scheduler is
```

File Name: NEPR_DSQ/CRAIG.BETA/NCB0600.FILE.14
Page : 00000000 0

The cursor, circled above, appears at the beginning of the file.

The Text Field

The right-hand column of the screen displays the printable ASCII (text) characters in the file.

```

00000000 73 65 64 2E 0A 54 68 65 20 66 69 72 73 74 20 72 sed..The first r
00000010 65 6C 65 61 73 65 20 6F 66 20 74 68 69 73 20 73 elease of this s
00000020 6F 66 74 77 61 72 65 20 77 69 6C 6C 20 75 73 65 oftware will use
00000030 20 65 74 68 65 72 6E 65 74 20 61 73 20 74 68 65 ethernet as the
00000040 0A 68 61 72 64 77 61 72 65 20 70 72 6F 74 6F 63 .hardware protoc
00000050 6F 6C 2E 0A 0A 49 6E 20 61 64 64 69 74 69 6F 6E ol...In addition
00000060 20 74 6F 20 74 68 65 20 6E 65 74 77 6F 72 6B 69 to the networki
00000070 6E 67 20 66 65 61 74 75 72 65 73 2C 20 62 75 67 ng features, bug
00000080 20 66 69 78 65 73 20 61 6E 64 20 65 6E 68 61 6E fixes and enhan
00000090 63 65 6D 65 6E 74 73 0A 73 69 6E 63 65 28 57 4D cements.since WM
000000A0 43 53 20 35 2E 30 20 61 72 65 20 69 6E 63 6F 72 CS 5.0 are incor
000000B0 70 6F 72 61 74 65 64 2E 20 20 4F 74 68 65 72 20 porated. Other
000000C0 69 6D 70 6F 72 74 61 6E 74 20 65 6E 68 61 6E 63 important enhanc
000000D0 65 6D 65 6E 74 73 20 74 6F 0A 57 4D 43 53 20 69 ements to.WMCS i
000000E0 6E 63 6C 75 64 65 3A 0A 0A 09 31 2E 20 41 20 6E nclude:...l. A n
000000F0 65 77 20 73 63 68 65 64 75 6C 65 72 20 69 73 20 ew scheduler is

```

File Name: MEMR.D00/CRABG.BETA/MCS0600.N.S.14
Page : 00000000 0

This column is the text field of the display. The text field is 16 characters wide. If a character in the file is not printable, such as a tab character or line feed, a period appears in its place in the text field.

General Information

The Hexadecimal Field

The four columns in the center of the screen display the hexadecimal values for the characters in the text field.

00000000	73	65	64	2E	0A	54	68	65	20	66	69	72	73	74	20	72	sed..The first r
00000010	65	6C	65	61	73	65	20	6F	66	20	74	68	69	73	20	73	elease of this s
00000020	6F	66	74	77	61	72	65	20	77	69	6C	6C	20	75	73	65	oftware will use
00000030	20	65	74	68	65	72	6E	65	74	20	61	73	20	74	68	65	ethernet as the
00000040	0A	68	61	72	64	77	61	72	65	20	70	72	6F	74	6F	63	.hardware protoc
00000050	6F	6C	2E	0A	0A	49	6E	20	61	64	64	69	74	69	6F	6E	ol...In addition
00000060	20	74	6F	20	74	68	65	20	6E	65	74	77	6F	72	6B	69	to the networki
00000070	6E	67	20	66	65	61	74	75	72	65	73	2C	20	62	75	67	ng features, bug
00000080	20	66	69	78	65	73	20	61	6E	64	20	65	6E	68	61	6E	fixes and enhan
00000090	63	65	6D	65	6E	74	73	0A	73	69	6E	63	65	20	57	4D	gements.since NM
000000A0	43	53	20	35	2E	30	20	61	72	65	20	69	6E	63	6F	72	CS 5.0 are incor
000000B0	70	6F	72	61	74	65	64	2E	20	20	4F	74	68	65	72	20	porated. Other
000000C0	69	6D	70	6F	72	74	61	6E	74	20	65	6E	68	61	6E	63	important enhanc
000000D0	65	6D	65	6E	74	73	20	74	6F	0A	57	4D	43	53	20	69	ements to.MACS i
000000E0	6E	63	6C	75	64	65	3A	0A	0A	09	31	2E	20	41	20	6E	nclude:...1. A n
000000F0	65	77	20	73	63	68	65	64	75	6C	65	72	20	69	73	20	ew scheduler is

File Name: MEPH_DSO/CRAIG.BETA/MCS0600.PLS.14
Page : 00000000 0

These columns make up the hexadecimal (hex) field of the display. For example, the 73 in the first position on the first line of the hex field is the hexadecimal equivalent of lowercase s, which you find in the corresponding position of the text field. The 0A in the fifth position on the first line of the hex field is the hexadecimal equivalent of the line-feed character, which is represented as a period in the corresponding position of the text field.

When you type or change a value in the hex field, Zap changes the corresponding character in the text field. Conversely, when you type or change a character in the text field, Zap changes the corresponding value in the hex field. Therefore, you can use Zap to edit either field.

Each hexadecimal value in the hex field represents one byte in the file.

General Information

Byte Addresses

The left-hand column of the screen displays the eight-digit address of the first byte in each line.

```
00000000 73 65 64 2E 0A 54 68 65 20 66 69 72 73 74 20 72 sed..The first r
00000010 65 6C 65 61 73 65 20 6F 66 20 74 68 69 73 20 73 elease of this s
00000020 6F 66 74 77 61 72 65 20 77 69 6C 6C 20 75 73 65 oftware will use
00000030 20 65 74 68 65 72 6E 65 74 20 61 73 20 74 68 65 ethernet as the
00000040 0A 68 61 72 64 77 61 72 65 20 70 72 6F 74 6F 63 .hardware protoc
00000050 6F 6C 2E 0A 0A 49 6E 20 61 64 64 69 74 69 6F 6E ol...In addition
00000060 20 74 6F 20 74 68 65 20 6E 65 74 77 6F 72 6B 69 to the networki
00000070 6E 67 20 66 65 61 74 75 72 65 73 2C 20 62 75 67 ng features, buy
00000080 20 66 69 78 65 73 20 61 6E 64 20 65 6E 68 61 6E fixes and enhan
00000090 63 65 6D 65 6E 74 73 0A 73 69 6E 63 65 20 57 4D cements.since NH
000000A0 43 53 20 35 2E 30 20 61 72 65 20 69 6E 63 6F 72 CS 5.0 are incor
000000B0 70 6F 72 61 74 65 64 2E 20 20 4F 74 68 65 72 20 porated. Other
000000C0 69 6D 70 6F 72 74 61 6E 74 20 65 6E 68 61 6E 63 important enhanc
000000D0 65 6D 65 6E 74 73 20 74 6F 0A 57 4D 43 53 20 69 ements to.WMCS i
000000E0 6E 63 6C 75 64 65 3A 0A 0A 09 31 2E 20 41 20 6E nclude:...l. A n
000000F0 65 77 20 73 63 68 65 64 75 6C 65 72 20 69 73 20 ew scheduler is
```

File Name: NEER_DEU/CRAIG.NEER/MCS0600.ELS.14

Page : 0000000 0

The numbers are for your convenience in keeping track of your position in the file. They represent the relative byte position, within the file, of the first hex value or character (i.e., the first byte) on each line. In the foregoing sample file, the 65 in the first position on the second line is byte number 10, the 6C is byte number 11, and so on to the 73 at the end of the line, which is byte number 1F in the file.

General Information

Page Numbers

The page numbers at the bottom of the screen are assigned by Zap for your convenience in moving about the file.

```
00000000 73 65 64 2E 0A 54 68 65 20 66 69 72 73 74 20 72 sed..The first r
00000010 65 6C 65 61 73 65 20 6F 66 20 74 68 69 73 20 73 elease of this s
00000020 6F 66 74 77 61 72 65 20 77 69 6C 6C 20 75 73 65 oftware will use
00000030 20 65 74 68 65 72 6E 65 74 20 61 73 20 74 68 65 ethernet as the
00000040 0A 68 61 72 64 77 61 72 65 20 70 72 6F 74 6F 63 .hardware protoc
00000050 6F 6C 2E 0A 0A 49 6E 20 61 64 64 69 74 69 6F 6E ol...In addition
00000060 20 74 6F 20 74 68 65 20 6E 65 74 77 6F 72 6E 69 to the networki
00000070 6E 67 20 66 65 61 74 75 72 65 73 2C 20 62 75 67 ng features, bug
00000080 20 66 69 78 65 73 20 61 6E 64 20 65 6E 68 61 6E fixes and enhanc
00000090 63 65 6D 65 6E 74 73 0A 73 69 6E 63 65 20 57 4D cements..since WM
000000A0 43 53 20 35 2E 30 20 61 72 65 20 69 6E 63 6F 72 CS 5.0 are incor
000000B0 70 6F 72 61 74 65 64 2E 20 20 4F 74 68 65 72 20 porated. Other
000000C0 69 6D 70 6F 72 74 61 6E 74 20 65 6E 68 61 6E 63 important enhanc
000000D0 65 6D 65 6E 74 73 20 74 6F 0A 57 4D 43 53 20 69 ements to.WMCS i
000000E0 6E 63 6C 75 64 65 3A 0A 0A 09 31 2E 20 41 20 6E nclude:...l. A n
000000F0 65 77 20 73 63 68 65 64 75 6C 65 72 20 69 73 20 ew scheduler is
```

File Name: WETA.DDD/CRAIG.BETA/MCS0600.PLS.14

Page : 00000000 0

Sixteen lines constitute one page (screenful) of data in Zap. Page numbers are given in both hexadecimal and decimal forms, respectively. For example, the first page in a file is 00000000 in the hexadecimal system, 0 in the decimal. The second is 00000001 and 1; the tenth, 0000000A and 10; and so on.

The Filename

The File Name entry at the bottom of the screen indicates the complete pathname for the file you are editing.

```
00000000 73 65 64 2E 0A 54 68 65 20 66 69 72 73 74 20 72 sed..The first r
00000010 65 6C 65 61 73 65 20 6F 66 20 74 68 69 73 20 73 elease of this s
00000020 6F 66 74 77 61 72 65 20 77 69 6C 6C 20 75 73 65 oftware will use
00000030 20 65 74 68 65 72 6E 65 74 20 61 73 20 74 68 65 ethernet as the
00000040 0A 68 61 72 64 77 61 72 65 20 70 72 6F 74 6F 63 .hardware protoc
00000050 6F 6C 2E 0A 0A 49 6E 20 61 64 64 69 74 69 6F 6E ol...In addition
00000060 20 74 6F 20 74 68 65 20 6E 65 74 77 6F 72 6B 69 to the networki
00000070 6E 67 20 66 65 61 74 75 72 65 73 2C 20 62 75 67 ng features, bug
00000080 20 66 69 78 65 73 20 61 6E 64 20 65 6E 68 61 6E fixes and enhan
00000090 63 65 6D 65 6E 74 73 0A 73 69 6E 63 65 20 57 4D cements.since VM
000000A0 43 53 20 35 2E 30 20 61 72 65 20 69 6E 63 6F 72 CS 5.0 are incor
000000B0 70 6F 72 61 74 65 64 2E 20 20 4F 74 68 65 72 20 porated. Other
000000C0 69 6D 70 6F 72 74 61 6E 74 20 65 6E 68 61 6E 63 important enhanc
000000D0 65 6D 65 6E 74 73 20 74 6F 0A 57 4D 43 53 20 69 ements to.WMCS i
000000E0 6E 63 6C 75 64 65 3A 0A 0A 09 31 2E 20 41 20 6E nclude....1. A n
000000F0 65 77 20 73 63 68 65 64 75 6C 65 72 20 69 73 20 ew scheduler is
```

File Name: ___MEPH_DS0/CRAIG.BETA/MCS0600.RLS.14

Page : 0000000 8

General Information

If the file is new, your terminal screen looks like the following example.

```
00000000 XX .....
00000010 XX .....
00000020 XX .....
00000030 XX .....
00000040 XX .....
00000050 XX .....
00000060 XX .....
00000070 XX .....
00000080 XX .....
00000090 XX .....
000000A0 XX .....
000000B0 XX .....
000000C0 XX .....
000000D0 XX .....
000000E0 XX .....
000000F0 XX .....

File Name:  _NEPH_DSU/SYSLIB/NEWFILE.DAT.1
Page      : 00000000 0
```

Whenever you are editing a file and there is no more data in the file, the rest of the hexadecimal display for the page is filled with lowercase x characters to indicate that there is no valid data from that point on in the file.

Editing a Disk Device

If you are directly editing a disk device, Zap displays sector numbers in addition to page numbers, as in the following example.

```

00000000 73 65 64 2E 0A 54 68 65 20 66 69 72 73 74 20 72 sed..The first r
00000010 65 6C 65 61 73 65 20 6E 66 20 74 68 69 73 20 73 elease of this s
00000020 6E 66 74 77 61 72 65 20 77 69 6C 6C 20 75 73 65 oftware will use
00000030 20 65 74 68 65 72 6E 65 74 20 61 73 20 74 68 65 ethernet as the
00000040 0A 68 61 72 64 77 61 72 65 20 70 72 6E 74 6E 63 .hardware protoc
00000050 6E 6C 2E 0A 0A 49 6E 20 61 64 64 69 74 69 6E 6E ol....In addition
00000060 20 74 6E 20 74 68 65 20 6E 65 74 77 6E 72 68 69 to the networki
00000070 6E 67 20 66 65 61 74 75 72 65 73 2C 20 62 75 67 ng features, bug
00000080 20 66 69 78 65 73 20 61 6E 64 20 65 6E 68 61 6E fixes and enhan
00000090 63 65 6D 65 6E 74 73 0A 73 69 6E 63 65 20 57 4D cements since WM
000000A0 43 53 20 35 2E 30 20 61 72 65 20 69 6E 63 6E 72 CS 5.0 are incor
000000B0 70 6E 72 61 74 65 64 2E 20 20 4E 74 68 65 72 20 porated. Other
000000C0 69 6D 70 6E 72 74 61 6E 74 20 65 6E 68 61 6E 63 important enhanc
000000D0 65 6D 65 6E 74 73 20 74 6E 0A 57 4D 43 53 20 69 ements to WMS i
000000E0 6E 63 6C 75 64 65 3A 0A 0A 09 31 2E 20 41 20 6E nclude....L. A n
000000F0 65 77 20 73 63 68 65 64 75 6C 65 72 20 69 73 20 ew scheduler is
    
```

Device Name: MEPH_DSO
Sector : 00000000
Page : 0

The sector and page numbers are displayed in hexadecimal form only.

Note that instead of a filename, Zap displays the devicename when you are editing a device.

The page number is not for the number of pages from the beginning of the disk; it is the page number in the sector you are editing. The number of pages per sector is determined by the sector size of the disk, divided by 256. For example, disks with 512-byte sectors have 2 pages per sector; disks with 1024-byte sectors have 4 pages per sector.

The byte addresses for a device are relative to the beginning of the sector being displayed.

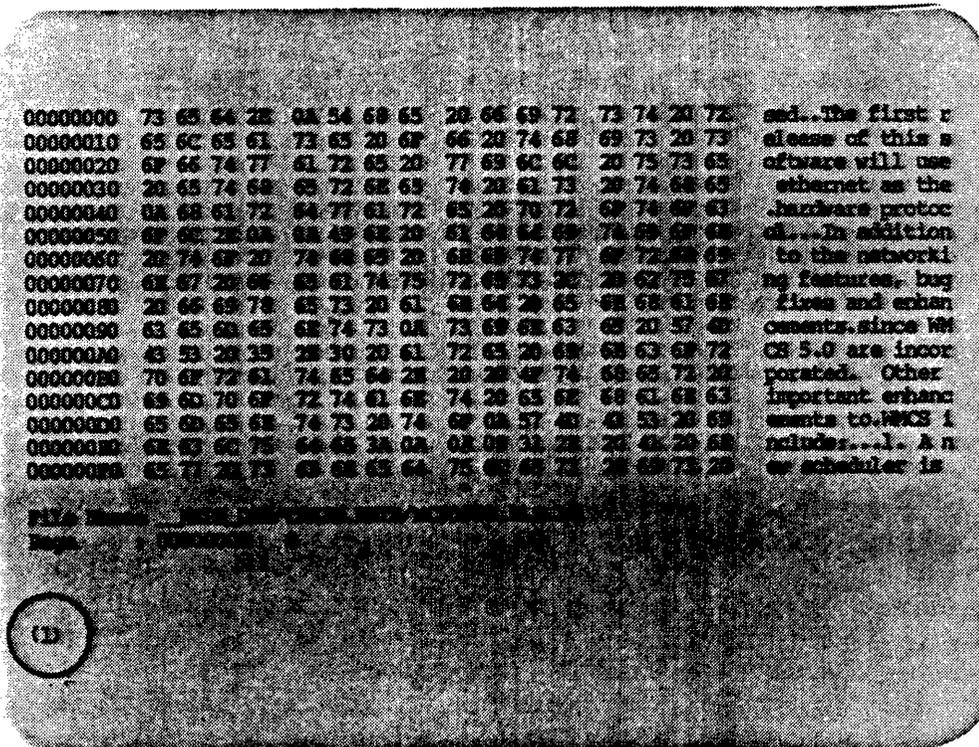
General Information

How to Execute Zap Functions

There are two ways to execute Zap functions: using the Zap function line and using single-keystroke commands. Generally speaking, single-keystroke commands perform minor changes quickly, whereas the function line performs more substantial changes. The method you use depends on what is more convenient for you and on your terminal type (single-keystroke commands vary according to terminal type).

The Zap Function Line

To call up the Zap function line, strike the escape key, [ESC], twice. The number 1 appears in parentheses at the bottom of the screen:



The bottom line of the screen, where the (1) appears, is the Zap function line. The number in parentheses represents the number of times this function will be repeated. It is called the function repetition number.

When you are on the Zap function line, type the two-letter mnemonic for the function you wish to execute. When you type the first letter of the mnemonic, the following prompt appears in place of the (1):

Cmd>

If the function requires no further input, it is executed immediately when you type the last letter of the mnemonic. If the function requires additional information, such as a page number or a filename, a prompt for that information appears when you type the last letter of the mnemonic. For example, the ef function calls the following prompt to the function line:

Zap File/Device name>

Type the information requested by the prompt. Then to execute the function, strike either [RETRN] or [ESC][ESC]. On the Zap function line, [RETRN] and [ESC][ESC] are equivalent.

If you enter the Zap function line and then decide not to perform a function, simply strike [ESC][ESC] again, or strike [RETRN]. You can also use the CANCEL command, [CTRL] c, to exit the function line at any time. That is, whether you are in the middle of typing a function or have just entered the function line, you can hold down the control key and type c to cancel the function line.

In this manual, Zap function-line commands are referred to by escape key and mnemonic: for example, [ESC][ESC] ef.

Single-keystroke Commands

Depending on the terminal type, certain Zap functions can be executed with a single keystroke, using either arrow keys, keypad keys, or control keys.

For details on using the alternate keypad, see the VEW User's Reference Manual. In this manual, function keys on the alternate keypad are enclosed in curly braces: for example, {4}.

To execute a control-key function, press the control key, [CTRL], and hold it down while you type the character for the command. The function is executed immediately, unless other input is required. If the function requires additional information, a prompt appears on the function line. For example, to move the cursor to the top of a page, hold down [CTRL] and type h. The cursor moves immediately. Most control-key functions can be repeated by keeping the control key depressed and repetitively striking the character key. In this manual, control-key functions are referred to by control key and character: for example, [CTRL] h.

General Information

Moving the Cursor

The cursor can be positioned in either the hex field or the text field on the screen. The cursor cannot be positioned between the hexadecimal characters in the hex field.

Use the arrow keys, the control key functions, and the function line commands to move the cursor.

If the cursor is located on the top or bottom margin of the screen, and the arrow key (or equivalent function-line command) that would move the cursor toward that same margin is pressed, the cursor jumps to the opposite margin in the same column. For example, if the cursor is in the first column at the bottom margin of the hex field and you strike the down-arrow, the cursor moves to the first column at the top margin of the hex field.

If the cursor is located on the right margin of the screen, pressing the right-arrow key moves the cursor to the left margin and down one line. If the cursor is on the bottom line, the cursor moves to the top line at the left margin.

If the cursor is located on the left margin of the screen, pressing the left-arrow key moves the cursor to the right margin and up one line. If the cursor is on the top line, the cursor moves to the bottom line at the right margin.

The arrow keys move the cursor from the hex field to the text field, but the word-movement function, ww, remains within whichever field of the screen the cursor was in when the function was executed. In other words, the only ways to switch between fields are to use the left- and right-arrow keys or to use the el and fl commands.

Making Changes

To change data in a file, whether the cursor is in the hex field or the text field, put the cursor on the character or digit you want to change and then type the new character or digit. The new character replaces (is typed over) the old one.

Help Display

To call up a list of Zap functions (a help display), strike [ESC][ESC] and type `he`.

The help display is several pages, or screenfuls, long. Once you call it to the screen, press any key to see the next page.

The help display is automatically exited after the last page is displayed. If you wish to exit the help display without viewing all the pages, type [CTRL] `c`. This returns you immediately to the file or device you were editing, and the cursor appears where it was before you called up the help display.

Unknown Functions

If you try to type on the Zap function line anything other than the mnemonics for the functions listed in this manual, a bell or beep sounds. The cursor remains on the function line. You can either type a recognized mnemonic or exit the function line.

The Minus Sign

Unless you specify otherwise, all Zap functions are performed toward the end of the file. However, the minus sign, `-`, can be used to perform some functions toward the beginning of the file.

For example, if you type `ml` onto the function line, the cursor moves one line toward the end of the page. But if you type `-ml` onto the function line, the cursor moves one line toward the beginning of the page. When you type the minus sign, the (1) on the function line changes to (-1).

Restoring Deleted Material

If you have modified or deleted the data in a page or sector and you wish to restore the page or sector to its original values, you can do so with the Undo function, `ud`. Read the description of this function in the Dictionary of Zap Commands, Chapter 3 of this manual.

General Information

Repeating and Editing the Function Line

Use [CTRL] \ to recall the previous function line. You can then edit it or execute it again. See the VEW User's Reference Manual for details on editing the function line.

Use [CTRL] z to execute the previous function without being able to edit it. For example, if the last function was the search function, [ESC][ESC] sr, you could use [CTRL] z to find the next occurrence of the same string.

File Versions

When Zap is invoked on an existing file, a new version of the file is not created. If any modifications are made with Zap and actually written to the disk, (see the sa and ef functions described in the Dictionary of Zap Functions, chapter 3 of this manual), the changes are made in the original file.

Diagnostic Messages

If you make an error in the Zap Program, a diagnostic message like the following appears at the bottom of your screen:

```
ZAP      : Opening file.txt
FAILED   : Status = 136.
MESSAGE  : The specified file is write-locked.
```

The first line of the diagnostic message tells you the name of the program (ZAP) and what the program was doing when the problem occurred. The second line tells you the status of the process (FAILED) and the diagnostic number (136). The third line is the message itself, explaining why the operation could not be completed.

General Information

If an error occurs that does not prohibit Zap from continuing execution, then the diagnostic message appears below the page number, as in the following example:

```
0000000 xx .....
0000010 xx .....
0000020 xx .....
0000030 xx .....
0000040 xx .....
0000050 xx .....
0000060 xx .....
0000070 xx .....
0000080 xx .....
0000090 xx .....
00000A0 xx .....
00000B0 xx .....
00000C0 xx .....
00000D0 xx .....
00000E0 xx .....
00000F0 xx .....

File Name:  WMSH_DSO/SYS/ID/MSMPLE.DSP.1
Page      : 0000000 0
Status    : The process tried to read past the logical end of a file.
```

A complete glossary of diagnostic messages appears in the WMCS User's Reference Manual.



Chapter 2

Directory of Zap Functions

In this directory, the functions are grouped according to what you use them to do: move the cursor, calculate a checksum, etc. Control-key functions and other functions, such as the alternate keypad and the arrow keys, are listed next to their equivalent escape-key functions. See Chapter 3 for an alphabetical list of Zap's escape-key functions. That chapter explains step-by-step how to execute each function.

The minus sign, -, in front of some of the escape-key functions means that this function is performed toward the beginning of the file.

The character keys specified in this chapter as control-key functions and alternate keypad functions are default character keys for these functions on a WICAT T7000 terminal. These character keys may differ according to your terminal type. If your terminal is not a T7000 or if you want to assign keys other than those specified here, see the VEW User's Reference Manual for further information.

Directory of Zap Functions

Move the Cursor to:

<u>Location</u>	<u>Escape-key functions</u>	<u>Control-key functions</u>	<u>Other functions</u>
Top of file	[ESC] [ESC] mt	[CTRL] t	
End of file	[ESC] [ESC] me	[CTRL] e	
Next Sector	[ESC] [ESC] ms		
Previous sector	[ESC] [ESC] -ms		
Next page	[ESC] [ESC] mp	[CTRL] p	
Previous page	[ESC] [ESC] -mp	[CTRL] h	
Specific sector or page	[ESC] [ESC] go	[CTRL] x	
Front of next line	[ESC] [ESC] ml	[CTRL] o	
Front of previous line	[ESC] [ESC] -ml	[CTRL] w	
Same column of next line	[ESC] [ESC] pl		down-arrow
Same column of previous line	[ESC] [ESC] -pl		up-arrow
Front of field on current line	[ESC] [ESC] fl	[CTRL] a	
End of field on current line	[ESC] [ESC] el	[CTRL] g	
Next word	[ESC] [ESC] mw	[CTRL] f	
Previous word	[ESC] [ESC] -mw	[CTRL] r	
Next character or byte	[ESC] [ESC] mc		right-arrow
Previous character or byte	[ESC] [ESC] -mc		left-arrow

Compute a checksum for:

<u>Type of block</u>	<u>Escape-key functions</u>	<u>Control-key functions</u>	<u>Other functions</u>
WMCS boot block	[ESC] [ESC] bc		
Other (non-WMCS) boot block	[ESC] [ESC] oc		
WMCS file control block (FCB)	[ESC] [ESC] fc		

Copy

<u>Item copied</u>	<u>Escape-key functions</u>	<u>Control-key functions</u>	<u>Other functions</u>
Page	[ESC] [ESC] co		
Sector	[ESC] [ESC] cs		

Directory of Zap Functions

Search

<u>Application</u>	<u>Escape-key functions</u>	<u>Control-key functions</u>	<u>Other functions</u>
Search (toward end of file)	[ESC] [ESC] sr		{7}
Search (toward top of file)	[ESC] [ESC] -sr		{4}

Create a CIP

<u>Application</u>	<u>Escape-key functions</u>	<u>Control-key functions</u>	<u>Other functions</u>
Create a CIP from Zap	[ESC] [ESC] cp		

Undo Changes

<u>Application</u>	<u>Escape-key functions</u>	<u>Control-key functions</u>	<u>Other functions</u>
Undo changes to page or sector	[ESC] [ESC] ud		{PF1}

Create a Print-file of:

<u>Item printed</u>	<u>Escape-key functions</u>	<u>Control-key functions</u>	<u>Other functions</u>
Current page	[ESC] [ESC] pr		

Set to a Constant Value

<u>Item set</u>	<u>Escape-key functions</u>	<u>Control-key functions</u>	<u>Other functions</u>
Page	[ESC] [ESC] sp		
Sector	[ESC] [ESC] ss		

Directory of Zap Functions

Screen Functions and Help Display

<u>Application</u>	<u>Escape-key functions</u>	<u>Control-key functions</u>	<u>Other functions</u>
Refresh screen	[ESC][ESC] rs	[CTRL] -	
Help display	[ESC][ESC] he		

File Control

<u>Application</u>	<u>Escape-key functions</u>	<u>Control-key functions</u>	<u>Other functions</u>
Abort Zap, do not save changes	[ESC][ESC] ab		
Save changes	[ESC][ESC] sa		{,}
Exit Zap and save changes	[ESC][ESC] ex		{-}
Edit new file or device	[ESC][ESC] ef	[CTRL] ^	

Toggle Functions

<u>Application</u>	<u>Escape-key functions</u>	<u>Control-key functions</u>	<u>Other functions</u>
Toggle mask8bit display flag	[ESC][ESC] tm		

Repeat or Edit the Previous Escape-key Function

<u>Application</u>	<u>Escape-key functions</u>	<u>Control-key functions</u>	<u>Other functions</u>
Repeat last function	[ESC][ESC] rp	[CTRL] z	
Recall last function line		[CTRL] \	

Control-characters on the Function Line

<u>Application</u>	<u>Escape-key functions</u>	<u>Control-key functions</u>	<u>Other functions</u>
Insert control-key character			{ENTER}

Directory of Zap Functions

Terminate a Zap Function

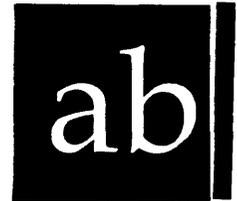
<u>Application</u>	<u>Escape-key functions</u>	<u>Control-key functions</u>	<u>Other functions</u>
Terminate Zap function		[CTRL] c	

Chapter 3

Dictionary of Zap Functions

This dictionary of escape-key functions is arranged alphabetically according to the mnemonics used to execute the functions.

Control-key functions and alternate keypad functions are described under their corresponding escape-key functions. Alternate keypad functions are indicated by curly braces, {}.



abort

Functional Description

Use this function to abort the current editing session; that is, to exit the current file without making any modifications.

Execution

Step 1 | Strike [ESC][ESC].

The number 1 appears at the bottom of your screen on the Zap function line.

Step 2 | Type ab.

If no modifications were made during this session of Zap, the file is immediately exited.

If modifications were made to the file, this prompt appears at the bottom of your screen:

File data has been modified - abort?

If you were editing a device, this prompt appears:

Device data has been modified - abort?

Step 3 | Type y.

This message appears at the bottom of your screen:

Aborting...

The screen is then cleared and the cursor appears on the CIP command line.

ab

Notes on Usage

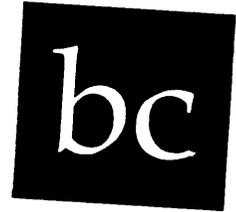
If you type a character other than y or n in response to the prompt, the following message appears momentarily:

 Please type 'Y' or 'N'.

The prompt then reappears.

Corresponding and Related Functions

[ESC][ESC] ex Exit file
[ESC][ESC] sa Save file



boot block checksum

Functional Description

Use this function to calculate the WMCS boot block checksum.

Execution

Step 1 | Strike [ESC][ESC].

The number 1 appears at the bottom of your screen on the Zap function line.

Step 2 | Type bc.

The new checksum is calculated and placed in the last four bytes of the display. The cursor reappears in the same position it was in prior to the execution of bc.

Notes on Usage

The bc function assumes the display is that of a disk boot block (disk boot blocks contain a checksum in the last four bytes).

Corresponding and Related Functions

[ESC][ESC] oc Calculate boot block checksum for other disks
[ESC][ESC] fc Calculate fcb checksum



copy

Functional Description

Use this function to copy a page from one part of the file or device to the current page.

Execution

For usage on a file:

- | | |
|--------|--|
| Step 1 | Strike [ESC][ESC].

The number 1 appears at the bottom of your screen on the Zap function line. |
| Step 2 | Type <code>co</code> .

A prompt like the following appears at the bottom of your screen:

Copy Page> |
| Step 3 | Type a page number.

This is the number of the page you wish to copy to the current page. The number may be either a decimal or hexadecimal value. Decimal values must begin with a percent sign, %. |
| Step 4 | Strike [RETRN].

The display refreshes as the current page is replaced by the copied page. The cursor reappears in the same position it was in prior to the execution of <code>co</code> . |

co

For usage on a device:

Step 1 | Strike [ESC][ESC].

The number 1 appears at the bottom of your screen on the Zap function line.

Step 2 | Type co.

A prompt like the following appears at the bottom of your screen:

Copy Sector>

Step 3 | Type a sector number.

The number may be either a decimal or hexadecimal value. Decimal values must begin with a percent sign, %.

For example, type 21.

Step 4 | Strike [RETRN].

An additional prompt like the following appears at the bottom of your screen:

Copy Sector> 21 Page>

Step 5 | Type a page number.

This is the number of the page you wish to copy to the current page. The number may be either a decimal or hexadecimal value. Decimal values must begin with a percent sign, %.

The valid range of page numbers on a disk device is determined by the sector size of the disk, divided by 256. For example, type either 0 or 1 for disks with 512-byte sectors, or type a number in the range 0-3 for disks with 1024-byte sectors.

Step 6 | Strike [RETRN].

The display refreshes as the current page is replaced by the copied page. The cursor reappears in the same position it was in prior to the execution of co.

Notes on Usage

If the page you specify is beyond the end of the file, the following message is briefly displayed:

The specified page is beyond the end of the file.

Corresponding and Related Functions

[ESC][ESC] cs Copy sector



cp

Functional Description

Use this function to go temporarily to the Command Interpreter Program (CIP) without losing your place in the file or closing or updating the file.

Execution

- | | |
|--------|---|
| Step 1 | Strike [ESC][ESC].

The number 1 appears at the bottom of your screen on the Zap function line. |
| Step 2 | Type <code>cp</code> .

The CIP prompt appears. |
| Step 3 | When you want to return to Zap, execute the CIP command <code>log</code> . See the <u>WMCS User's Reference Manual</u> for a complete explanation of the <code>log</code> command.

When you return to Zap, the cursor reappears in the same position it was in prior to the execution of <code>cp</code> . |

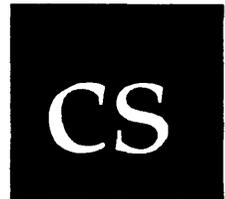
Notes on Usage

None.

cp

Corresponding and Related Functions

[ESC] c Create a CIP



copy sector

Functional Description

Use this function to copy a sector from one part of the file or device to the current sector.

Execution

- | | |
|--------|--|
| Step 1 | Strike [ESC][ESC].

The number 1 appears at the bottom of your screen on the Zap function line. |
| Step 2 | Type <code>cs</code> .

The following prompt appears on the function line:

Copy Sector> |
| Step 3 | Type the number of the sector you wish to copy to the current sector.

The number may be either a decimal or hexadecimal value. Decimal values must begin with a percent sign, %. |
| Step 4 | Strike [RETRN].

The display refreshes, showing the contents of the newly copied sector. The cursor reappears in the same position it was in prior to the execution of <code>cs</code> . |

CS

Notes on Usage

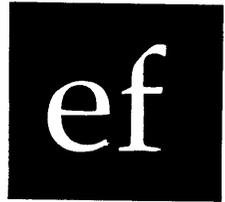
If the sector you specify in step 3 is beyond the end of the file, the following message is briefly displayed:

The specified sector is beyond the end of the file.

A sector is the same size as a page when you are editing a file; that is, cs functions the same as co when you are editing a file.

Corresponding and Related Functions

[ESC][ESC] co Copy page



edit file

Functional Description

Use this function to edit a new file without exiting Zap.

Execution

Step 1 Strike [ESC][ESC].

The number 1 appears at the bottom of your screen on the Zap function line.

Step 2 Type `ef`.

A prompt like this appears at the bottom of your screen:

```
Zap File/Device name> FILE.TXT
```

The cursor appears on the first letter of the filename you are now editing, `FILE.TXT` in the example above.

Step 3 Type a filename, or a devicename preceded by an underscore.

ef

- Step 4 | Strike [RETRN].
- If you have not modified any data in the current file, the current file is aborted and the new file you specified appears on the screen.
- If you have made any modifications to the file you are currently editing, the following message appears at the bottom of your screen:
- File data has been modified - save?
- Or if you have made modifications to the device you are editing, this message appears:
- Device data has been modified - save?
- When one of the foregoing messages appears, go to step 5.
- Step 5 | If you want to save your modifications, type y. If you do not want to save them, type n.
- If you type y, the current file modifications are saved, and then the new file appears on the screen.
- If you type n, the current file is aborted, and the new file appears on the screen.
- Should you respond with any character other than y or n, the following message appears briefly on your screen:
- Please type 'Y' or 'N'.
- Then the message from step 4 reappears on the screen.

Notes on Usage

If the new file does not exist, it is created.

Corresponding and Related Functions

[CTRL] ^ Edit another file or device



end of line

Functional Description

Use this function to move the cursor to the end of the current line in either the hex or the text field, or to change fields on the current line.

Execution

- | | |
|--------|---|
| Step 1 | Strike [ESC][ESC].

The number 1 appears at the bottom of your screen on the Zap function line. |
| Step 2 | Type <code>el</code> .

The cursor moves to the end of the current field. If it is already at the end of a field, the cursor moves to the front of the other field on the line. |

Notes on Usage

None.

Corresponding and Related Functions

[CTRL] g Move cursor to end of line
[ESC][ESC] fl Move cursor to front of line



exit

Functional Description

Use this function to exit the file you are editing and simultaneously save any modifications you have made.

Execution

- | | |
|--------|---|
| Step 1 | Strike [ESC][ESC].

The number 1 appears at the bottom of your screen on the Zap function line. |
| Step 2 | Type ex .

This kind of message appears at the bottom of your screen:

Exiting FILE.TXT

Then the screen is cleared and the cursor appears next to the CIP prompt (right angle bracket) on the CIP command line. |

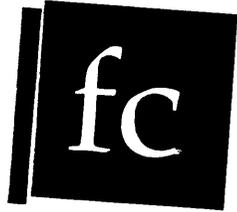
Notes on Usage

None.

ex

Corresponding and Related Functions

{-}	Exit file
[ESC][ESC] ab	Abort file
[ESC][ESC] sa	Save modifications to file



fc checksum

Functional Description

Use this function to calculate the FCB checksum.

Execution

Step 1 | Strike [ESC][ESC].

The number 1 appears at the bottom of your screen on the Zap function line.

Step 2 | Type `fc`.

The new checksum is calculated and placed in the last two bytes of the display. The cursor reappears in the same position it was in prior to the execution of `fc`.

Notes on Usage

The `fc` function assumes the display is that of an FCB entry in the file named `/ROOTDIR/FCB.SYS` (FCB entries contain a checksum in the last two bytes).

Corresponding and Related Functions

[ESC][ESC] bc Calculate boot block checksum for WMCS disks
[ESC][ESC] oc Calculate boot block checksum for other disks



front of line

Functional Description

Use this function to move the cursor to the front of the current line in either the hex or the text field, or to change fields on the current line.

Execution

- | | |
|--------|---|
| Step 1 | Strike [ESC][ESC].

The number 1 appears at the bottom of your screen on the Zap function line. |
| Step 2 | Type fl.

The cursor moves to the front of the current field. If it is already at the front of the field, the cursor moves to the end of the other field on the line. |

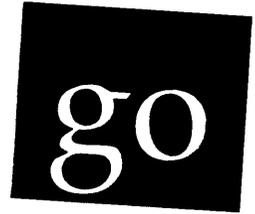
Notes on Usage

None.

f1

Corresponding and Related Functions

[CTRL] a Move cursor to front of line
[ESC][ESC] e1 Move cursor to end of line



go to sector/page

Functional Description

Use this function to go to a specific page number in a file, or to a specific sector and page number on a device.

Execution

For usage on a file:

Step 1 | Strike [ESC][ESC].

The number 1 appears at the bottom of your screen on the Zap function line.

Step 2 | Type go.

A prompt like the following appears at the bottom of your screen:

Go to Page>

Step 3 | Type a page number.

The number may be either a decimal or hexadecimal value. Decimal values must begin with a percent sign, %.

Step 4 | Strike [RETRN].

The screen is repainted with the new page of data.

go

For usage on a device:

- Step 1 | Strike [ESC][ESC].
- The number 1 appears at the bottom of your screen on the Zap function line.
- Step 2 | Type go.
- A prompt like the following appears at the bottom of your screen:
- Go to Sector>
- Step 3 | Type a sector number.
- The number may be either a decimal or hexadecimal value. Decimal values must begin with a percent sign, %.
- For example, type 21.
- Step 4 | Strike [RETRN].
- An additional prompt like the following appears at the bottom of your screen:
- Go to Sector> 21 Page>
- Step 5 | Type a page number.
- The valid range of page numbers on a disk device is determined by the sector size of the disk, divided by 256. For example, type either 0 or 1 for disks with 512-byte sectors, or type a number in the range 0-3 for disks with 1024-byte sectors.
- Step 6 | Strike [RETRN].
- The screen is repainted with the new page of data.

Notes on Usage

If you type a number that is not in the valid range for the given device, a message like the following appears briefly at the bottom of your screen:

The page number must be between 0 and 0003.

Then the cursor returns to the place on the current page where it was before the `go` function was initiated.

Corresponding and Related Functions

[CTRL] x	Move to specific page or sector
[ESC][ESC] me	Move to end of file or device
[ESC][ESC] mt	Move to top of file or device



help

Functional Description

Use this function to display the help screens for Zap.

Execution

Step 1 | Strike [ESC][ESC].

The number 1 appears at the bottom of your screen on the Zap function line.

Step 2 | Type `he`.

The data on the screen are erased, and the first screen of the help display is shown. Near the bottom of the screen the following prompt appears:

Press any key to continue . . .

There are several pages in the help display. After the last page is displayed, the data from the current page is redisplayed, and the cursor reappears in the same position it was in prior to the execution of `he`.

Notes on Usage

You may exit the help display without going through all the pages by typing [CTRL] c.

he

Corresponding and Related Functions

None.



move character

Functional Description

Use this function to move the cursor one character forward or backward.

Execution

Step 1 | Strike [ESC][ESC].

The number 1 appears at the bottom of your screen on the Zap function line.

Step 2 | Type `mc` to move the cursor to the next hex position (if the cursor is in the hex field) or to the next text position (if the cursor is in the text field), or type `-mc` to move the cursor to the previous hex or text position.

If the cursor is in the last text position on a line when you execute `mc`, it jumps to the first hex position on the next line. If the cursor is in the last text position on the last line of the screen, it moves to the first hex position on the first line of the screen. If the cursor is in the last hex position on a line, it moves to the first text position on the same line.

If the cursor is in the first hex position on a line when you execute `-mc`, it jumps to the last text position on the previous line. If the cursor is in the first hex position on the first line of the screen, it moves to the last text position on the last line of the screen. If the cursor is in the first text position on a line, it moves to the last hex position on the same line.

mc

Notes on Usage

None.

Corresponding and Related Functions

right-arrow	Move cursor to next character
left-arrow	Move cursor to previous character



move to end

Functional Description

Use this function to move the cursor to the end of the current file or device.

Execution

Step 1 | Strike [ESC][ESC].

The number 1 appears at the bottom of your screen on the Zap function line.

Step 2 | Type *me*.

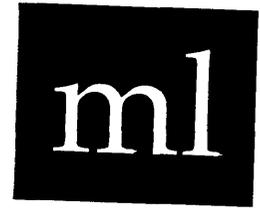
The cursor moves to the last position in the hex field on the last page (if the cursor is in the hex field), or to the last position in the text field on the last page (if the cursor is in the text field).

Notes on Usage

None.

Corresponding and Related Functions

[CTRL] e Move to end of current file or device
[ESC][ESC] mt Move to top of current file or device



move line

Functional Description

Use this function to move the cursor to the front of the next or previous line.

Execution

Step 1 | Strike [ESC][ESC].

The number 1 appears at the bottom of your screen on the Zap function line.

Step 2 | Type `ml` to move the cursor to the front of the next line in the same field, or type `-ml` to move the cursor to the front of the previous line in the same field.

If the cursor is on the last line of the screen (in either field) when you type `ml`, the cursor moves to the front of the first line.

If the cursor is on the first line of the screen (in either field) when you type `-ml`, the cursor moves to the front of the last line.

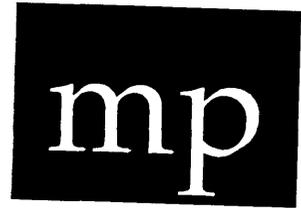
Notes on Usage

None.

ml

Corresponding and Related Functions

[CTRL] o Move cursor to front of next line
[CTRL] w Move cursor to front of previous line
[ESC][ESC] pl Move cursor up or down a line in same column



move page

Functional Description

Use this function to page forward or backward through the file.

Execution

Step 1 Strike [ESC][ESC].

The number 1 appears at the bottom of your screen on the Zap function line.

Step 2 Type `mp` to move to the next page, or type `-mp` to move to the previous page.

If the cursor is not on the last line of the display when you type `mp`, it moves to the last line. Once the cursor is on the last line, `mp` refreshes the screen with the next page of data. The cursor reappears in the last position of the hex field (if it was in the hex field when you executed `mp`) or in the last position of the text field (if it was in the text field when you executed `mp`).

If the cursor is not on the first line of the display when you type `-mp`, it moves to the first line. Once the cursor is on the first line, `-mp` refreshes the screen with the previous page of data. The cursor appears in the first position of the hex field or the first position of the text field, whichever it was in before you executed `-mp`.

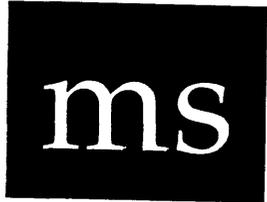
mp

Notes on Usage

None.

Corresponding and Related Functions

[CTRL] p	Move cursor to next page
[CTRL] h	Move cursor to previous page
[ESC][ESC] go	Go to a specific page
[ESC][ESC] ms	Move to the next or previous sector of file or device



move sector

Functional Description

Use this function to move to the next (or previous) sector on a device.

Execution

- | | |
|--------|---|
| Step 1 | Strike [ESC][ESC]. |
| | The number 1 appears at the bottom of your screen on the Zap function line. |
| Step 2 | Type <code>ms</code> to move the cursor to the next sector, or type <code>-ms</code> to move the cursor to the previous sector. |
| | The screen is refreshed with the next (or previous) sector of data. The cursor reappears in the same position it had on the page from which it was moved. |

Notes on Usage

This function applies to devices only. If executed on a file, it has the same effect as the `mp` function.

The `ms` function always moves to page 0 within the specified sector, regardless of the page you are on when you execute `ms`.

Corresponding and Related Functions

[ESC][ESC] go Go to a particular page
[ESC][ESC] me Move to the last page of the file or device
[ESC][ESC] mp Move to the next or previous page of the file or device
[ESC][ESC] mt Move to the the first page of the file or device



move to top

Functional Description

Use this function to move the cursor to the top of the file or device.

Execution

Step 1 | Strike [ESC][ESC].

The number 1 appears at the bottom of your screen on the Zap function line.

Step 2 | Type mt.

The cursor moves to the first position in the hex field on the first page (if the cursor is in the hex field), or the first position in the text field on the first page (if the cursor is in the text field).

Notes on Usage

None.

Corresponding and Related Functions

[CTRL] t Move to top of file or device
[ESC][ESC] me Move to end of file or device



move word

Functional Description

Use this function to move the cursor to the beginning of the next word or to the end of the previous word.

Execution

- | | |
|--------|--|
| Step 1 | Strike [ESC][ESC].

The number 1 appears at the bottom of your screen on the Zap function line. |
| Step 2 | Type <code>mw</code> to move the cursor to the beginning of the next word, or type <code>-mw</code> to move the cursor to the end of the previous word.

If the cursor is at the last word on a line when you type <code>mw</code> , it moves to the first word on the next line. If the cursor is at the last word on the last line, it moves to the first word on the first line.

If the cursor is at the first word on the line when you type <code>-mw</code> , it moves to the last word on the previous line. If the cursor is at the first word on the first line, it moves to the last word on the last line. |

Notes on Usage

This function considers alphanumeric characters as words in the text field. In the hex field, this function moves from byte to byte.

Corresponding and Related Functions

[CTRL] f Move cursor to next word
[CTRL] r Move cursor to previous word



other checksum

Functional Description

Use this function to calculate the checksum for other (non-WMCS) boot blocks.

Execution

- | | |
|--------|--|
| Step 1 | Strike [ESC][ESC].

The number 1 appears at the bottom of your screen on the Zap function line. |
| Step 2 | Type <code>oc</code> .

The new checksum is calculated and placed in the last four bytes of the display. The cursor reappears in the same position it was in prior to the execution of <code>oc</code> . |

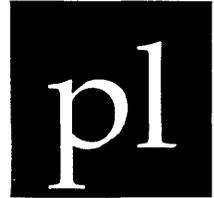
Notes on Usage

The `oc` function assumes the display is that of a disk boot block for an operating system other than WMCS, e.g., for UNIX (disk boot blocks contain a checksum in the last four bytes).

oc

Corresponding and Related Functions

[ESC][ESC] bc Calculate boot block checksum for WMCS disks
[ESC][ESC] fc Calculate fcb checksum



position in line

Functional Description

Use this function to move the cursor up or down a line in the same column.

Execution

- | | |
|--------|--|
| Step 1 | Strike [ESC][ESC].

The number 1 appears at the bottom of your screen on the Zap function line. |
| Step 2 | Type pl to move the cursor down a line on the screen, or type -pl to move the cursor up a line.

If the cursor is on the last line when you type pl, it moves to the same column on the first line.

If the cursor is on the first line when you type -pl, it moves to the same column on the last line. |

Notes on Usage

None.

pl

Corresponding and Related Functions

up-arrow	Move cursor up a line
down-arrow	Move cursor down a line
[ESC][ESC] ml	Move cursor to front of next or previous line



print

Functional Description

Use this function to create a print file of the current page.

Execution

Step 1 | Strike [ESC][ESC].

The number 1 appears at the bottom of your screen on the Zap function line.

Step 2 | Type pr.

The following message appears briefly at the bottom of the screen:

Printing page...

Notes on Usage

Zap creates in the current directory a new file named ZAP.PRN, which is a copy of the data on the screen at the time the pr function is executed. A time stamp is appended to the end of the file. You can print ZAP.PRN with the CIP command print.

pr

Corresponding and Related Functions

None.



repeat function

Functional Description

Use this function to repeat the last function you executed.

Execution

- | | |
|--------|---|
| Step 1 | Strike [ESC][ESC].

The number 1 appears at the bottom of your screen on the Zap function line. |
| Step 2 | Type rp.

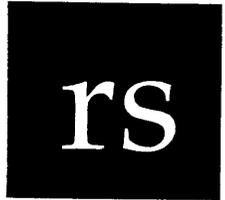
The last function you initiated from the Zap function line is executed. |

Notes on Usage

If no previous function exists, a bell or beep is sounded.

Corresponding and Related Functions

- | | |
|----------|-------------------------------|
| [CTRL] z | Repeat the last function |
| [CTRL] \ | Recall the last function line |



refresh screen

Functional Description

Use this function to redisplay (refresh) the screen.

Execution

Step 1 | Strike [ESC][ESC].

The number 1 appears at the bottom of your screen on the Zap function line.

Step 2 | Type `rs`.

The screen is redisplayed, and the cursor returns to the position it had prior to the `rs` function.

Notes on Usage

Use `rs` to restore the display on the screen when the display is jumbled. For example, when a user sends you a message and you are in Zap, that message appears on your screen and disrupts the display. The `rs` function restores the display to its condition before the interruption and returns the cursor to its original position.

Corresponding and Related Functions

[CTRL] - Redisplay the screen



save changes

Functional Description

Use this function to save all the modifications made to the file or device so far.

Execution

- | | |
|--------|---|
| Step 1 | Strike [ESC][ESC]. |
| | The number 1 appears at the bottom of your screen on the Zap function line. |
| Step 2 | Type <u>sa</u> . |
| | The following message is briefly displayed at the bottom of your screen: |
| | Saving modifications... |
| | The cursor returns to the position it had prior to the <u>sa</u> function. |

Notes on Usage

Use sa to periodically save modifications to a file or device, if you are certain that you want the changes made and that you will not be exiting Zap shortly. This makes it possible to save the work you have done so that in case the machine crashes, the modifications made prior to the last save will have been written to the disk.

If no modifications have been made, this function has no effect.

sa

Corresponding and Related Functions

{,}	Save modifications
[ESC][ESC] ab	Abort file
[ESC][ESC] ex	Exit Zap



set page

Functional Description

Use this function to set the page to a constant value.

Execution

- | | |
|--------|--|
| Step 1 | Strike [ESC][ESC].

The number 1 appears at the bottom of your screen on the Zap function line. |
| Step 2 | Type <i>sp</i> .

A prompt like the following appears at the bottom of your screen:

Set Page to Constant> |
| Step 3 | Type a number.

The number may be either a decimal or hexadecimal value. Decimal values must begin with a percent sign, %.

For example if you type 9E in response to the prompt, the entire page (or to the end of the valid data on the page) will be set to the hex pattern 9E. |
| Step 4 | The cursor returns to the place on the page where it was before the function was issued. |

sp

Notes on Usage

None.

Corresponding and Related Functions

[ESC][ESC] ss Set sector to constant

The logo consists of the letters 'sr' in a white, serif font, centered within a solid black square.

search

Functional Description

Use this function to search for a hexadecimal or character string.

Execution

Step 1 | Strike [ESC][ESC].

The number 1 appears at the bottom of your screen on the Zap function line.

Step 2 | Type `sr` to search forward through the file, or type `-sr` to search backward.

If you type `sr`, a prompt like the following appears at the bottom of your screen:

Search Fwd>

If you type `-sr`, the following prompt appears at the bottom of your screen:

Search Bwd>

sr

- Step 3 | Type the string to be searched for.
- You can enter the string in hexadecimal bytes. For example, you could type the following:
- 4e 75 4e 5a ff fc
- Or you can enter the string in text characters. When you search for text characters, they must be enclosed in either single or double quotation marks. For example, you could type the following:
- "find this string"**
- NOTE: You must enclose the character string in identical quotation marks. That is, do not begin the string with a single quotation mark and end it with a double quotation mark, or vice versa.
- Step 4 | Strike [RETRN].
- Zap begins searching for the string. If the string is found, the cursor reappears at the end of the string, in whichever field the cursor was in when you executed sr.
- If the string is not found, the following message appears briefly at the bottom of the screen:
- Search argument not found.
- The cursor then returns to the point where it was prior to the execution of the sr function.

Notes on Usage

If you don't enter any data on the function line, or if you enter invalid data when the cursor is in the hex field, one of the following error messages appears briefly at the bottom of the screen:

The search argument must have at least one byte of data to search for.

An invalid character appears in a decimal string.

No number was found during a search or scan for a number.

Corresponding and Related Functions

{7} Search forward
{4} Search backward



SS

set sector

Functional Description

Use this function to set the sector to a constant value.

Execution

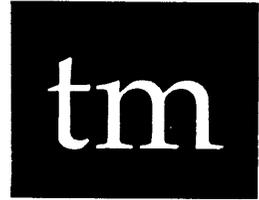
- Step 1 | Strike [ESC][ESC].
- The number 1 appears at the bottom of your screen on the Zap function line.
- Step 2 | Type `ss`.
- A prompt like the following appears at the bottom of your screen:
- Set Sector to Constant>
- Step 3 | Type a number.
- The number may be either a decimal or hexadecimal value. Decimal values must begin with a percent sign, %.
- For example if you type 9E in response to the prompt, the entire sector (or to the end of the valid data in the sector) will be set to the hex pattern 9E.
- Step 4 | The cursor returns to the place on the page where it was before the function was executed.

Notes on Usage

None.

Corresponding and Related Functions

[ESC][ESC] sp Set page to constant



toggle mask8bit

Functional Description

Use this function to enable or disable display of characters with bit 8 set.

Execution

Step 1 | Strike [ESC][ESC].

The number 1 appears at the bottom of your screen on the Zap function line.

Step 2 | Type `tm`.

The screen is updated to reflect any changes in the ASCII section (text field) of the display based on the state of the `mask8bit` flag, and the cursor returns to the position it had prior to the `tm` function.

Notes on Usage

Normally, Zap does not display characters with bit 8 set, since these characters are outside of the ASCII character set which is defined for characters whose values range from 0 to 127 (\$0 to \$7F). The `tm` function toggles the state of the `mask8bit` flag. By default, the flag is not set and these characters are not displayed. The state of this flag can be affected by the `:mask8bit` command-line switch and by the `tm` function.

tm

Corresponding and Related Functions

None.



undo changes

Functional Description

Use this function to undo changes.

Execution

- | | |
|--------|--|
| Step 1 | Strike [ESC][ESC]. |
| | The number 1 appears at the bottom of your screen on the Zap function line. |
| Step 2 | Type ud. |
| | The screen is redisplayed, showing the data that was originally in the file (or sector on the device), and the cursor returns to the position it had prior to the ud function. |

Notes on Usage

This function affects all changes made to a particular page or sector, even if you have left that page or sector and returned again to make modifications any number of times, (unless you have saved the modifications explicitly with the sa function, or unless Zap requested a confirmation of a save because there were no more buffers to hold modified data).

ud

Corresponding and Related Functions

{PF1} (on a T7000) Undo changes
{PF0} (on a VISUAL200) Undo changes

WICAT Systems, Inc.

Product-documentation Comment Form

We are constantly improving our documentation, and we welcome specific comments on this manual.

Document Title: _____

Part Number: _____

- Your Position:**
- | | |
|--|--|
| <input type="checkbox"/> Novice user | <input type="checkbox"/> System manager |
| <input type="checkbox"/> Experienced user | <input type="checkbox"/> Systems analyst |
| <input type="checkbox"/> Applications programmer | <input type="checkbox"/> Hardware technician |

Questions and Comments

Page No.

Briefly describe examples, illustrations, or information that you think should be added to this manual.

_____	_____
_____	_____
_____	_____

What would you delete from the manual and why?

_____	_____
_____	_____
_____	_____

What areas need greater emphasis?

_____	_____
_____	_____
_____	_____

List any terms or symbols used incorrectly.

_____	_____
_____	_____
_____	_____

First Fold

	<p>NO POSTAGE NECESSARY IF MAILED IN THE UNITED STATES</p> 
<p>BUSINESS REPLY MAIL FIRST CLASS PERMIT NO. 00028 OREM, UTAH</p>	
<p>POSTAGE WILL BE PAID BY ADDRESSEE</p>	
<p>WICAT Systems, Inc. Attn: Corporate Communications 1875 S. State St. Orem, UT 84058</p>	

Second Fold

Tape