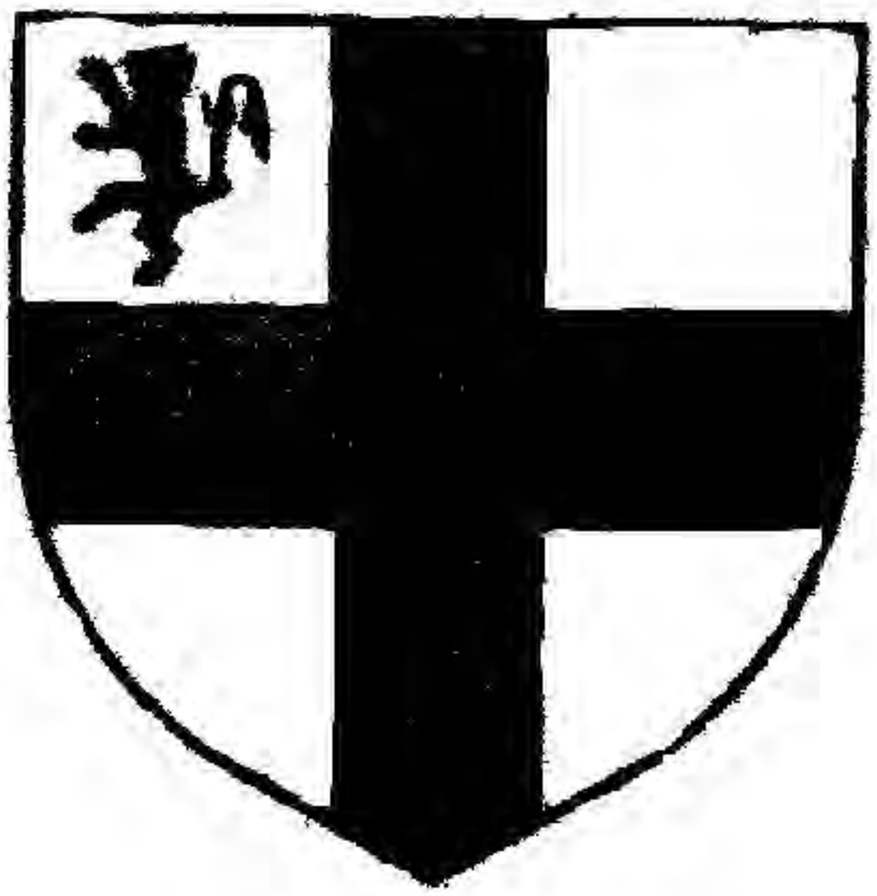
EZGen

Boot File Editor for OS9
Version 1.0B9



P.O. Box 58342
Renton, WA 98058

EZGen

Syntax: ezgen [opts] file

Function: Edit one or more OS9 modules or data blocks contained in a specified file.

Parameters:

file The name of the file to be edited.

opts Any combination of the following options may be used to modify the operation of EZGen:

-? -d -f -i -k -m -o

The operation of individual options is explained below. Options can be run together (e.g. -ru is equivalent to -r -u).

Options:

-? Display options summary message.

-d Descriptive mode. This option causes EZGen to display status information and interactive prompts.

-f Permit fragmented output file.

-i No input — create output file immediately.

-k Do not move or change the size of the output file.

-m Manual mode — do not abort after command errors.

-o Do not generate an output file.

Commands:

? Display a list of the size, approximate loading offset, and name of each module in the file. The loading offset is used only with boot files. It indicates what the module's approximate memory offset would be if the file were booted. This can be used as an aid in arranging the boot file to force modules to particular offsets.

$cmd Fork shell and execute command 'cmd'

Display name and size of module at current module pointer. The size is displayed in hexadecimal.

PUBLISHED BY BURKE & BURKE
P.O. BOX 58342
RENTON, WA 98058

LIMITED WARRANTY

Burke & Burke warrants the PRODUCT against defects in material or workmanship for a period of ninety (90) days from the date of purchase by the original owner. This warranty is limited to repair or replacement of PRODUCT which proves defective during this period, at the sole expense and discretion of Burke & Burke. This warranty specifically excludes software defects and defects caused by negligence, abuse, accident, or tampering.

DISCLAIMER

Every effort has been made to ensure the accuracy of this manual and the quality of the PRODUCT it describes. Burke & Burke makes no warranties, whether expressed, statutory, or implied, of any kind whatsoever, as to the merchantability of the PRODUCT or it's fitness for a particular use, except as set forth above as the LIMITED WARRANTY.

Burke & Burke does not assume any liability for any damages, whether special, indirect, or consequential, resulting from the use of the PRODUCT. The PRODUCT is sold on an AS-IS basis.

```
*         Treat entire input line as a comment.

b         Move module pointer to end of file.

c x z     Change value of byte at offset 'x' in current module
          to 'z'.  THIS COMMAND DOES NOT UPDATE THE MODULE CRC.

c x y z   Change value of byte at offset 'x' in current module
          from 'y' to 'z'.  An error is generated if the byte
          does not match 'y'.  THIS COMMAND DOES NOT UPDATE THE
          MODULE CRC.

d         Delete current module from file and set module pointer
          to the next module.

f         Display name of file being edited.

g name    Get module from memory.  The specified module is copied
          from memory to the workfile, at the location specified
          by the module pointer.  The module pointer is moved to
          the copied module.

h         Add module header, module name and CRC to data block
          at current module pointer position.  This command will
          convert a block of data into an OS9 module.  One
          application of this command is to convert an image of
          an RS-DOS binary file to a format the can be examined
          by an OS9 disassembler.

i path    Insert contents of file 'path' before current module,
          and move module pointer to first inserted module or
          data block.

l name    Move module pointer to specified module or data block.

m x       Display the hex/ASCII data stored at the 16 consecutive
          module offsets, beginning at offset 'x'.

m x y     Display the hex/ASCII data stored at the range of module
          offsets beginning at offset 'x' and ending at offset 'y'.

n         Move module pointer to next module or data block.

p path    Patch current module using data in file 'path'.  The
          data in 'path' must be stored in either Burke & Burke
          PATCH file format or RS-DOS BINARY file format.  The
          module CRC is corrected automatically.

q         Create output file and quit.

r name    Change the name of the current module.  If the new name
          is longer than the old name, the module is extended and
          the new name is placed at the end of the module.  The
          module CRC is corrected automatically.

r n x     Change a name stored in the current module.  The 16-bit
          value stored at module offset 'x' is used as the module
          offset to the name string.  The name is changed to 'n'
```

* Data or modules in the output file will be stored in consecutively numbered disk sectors if no options are given. The options change the format of the output file as follows:

Options	Output File Generation
none	1) Rename input file by appending "$2" to its name. 2) Create contiguous copy of work file with same name as original input file. 3) Delete work file. 4) Delete renamed input file.
-f	1) Delete input file. 2) Rename work file to name of original input file.
-k	1) Verify that work file has same length as input file. 2) Copy contents of work file over old contents of input file. 3) Delete work file. 4) Delete renamed input file.
-o	1) Exit without updating input file or deleting work file.

* EZGen can be used on any file. Not all files contain OS9 modules, and certain special files contain both data and OS9 modules. EZGen uses the OS9 "sync byte" convention to detect the start of a module. Any part of the file that does not follow the "sync byte" convention is assumed to be a data block, which ends when the next set of sync bytes are detected. A file that does not contain any sync bytes (such as a text file) will be treated as a single data block.

  The sync byte pattern is $87CD.

* The -F option should be used when editing files other than the kernal, OS9BOOT and ALTBOOT. This option writes a standard OS9 output file instead of a special contiguous output file.

* Use the -D and -H options when entering EZGen commands from the keyboard.

* EZgen automatically assigns temporary names to data blocks. These names have the form:

  tmp?

  The relative position of the data block is used to replace '?' in the name. For example, the first data block in a file is named 'tmp0'; the fourth data block is named 'tmp3'; and so on.

The names of data blocks can change any time a data block is added or deleted.

* Burke & Burke PATCH format is illustrated below. All values are stored as 8-bit binary codes.

Data Record
---

00 <size16> <offset16> <data bytes>

There may be any number of data records. The number of data bytes in the record is given by the 2 byte number, <size16>. There must be exactly <size16> data bytes. The data bytes will be used to replace data in the current module beginning at the offset given by the 2 byte number, <offset16>.

EOF Record

FF 00 00 00 00

Exactly one EOF record must be present at the end of the file.

* The module header created by the "M" command specifies a module type of "6809 object code program". The name follows the 13 byte module header immediately. EZGen always allocates 33 bytes for the module name. The execution offset is set to the 1st byte of the data block. A correct CRC is appended to the data block.

Examples:

* To install a new version of the CC3DISK device driver in the bootfile on drive /d0:

```
OS9:ezgen /d0/os9boot

EZGen Version 1.08
Copyright 1990 Burke & Burke

*** Link to old module
l cc3disk
*** Replace with new module from /d1
u /d1/modules/cc3disk.dr
*** All done.
q

OS9:
```

* To change the name of a floppy drive from "/d0" to "/f0":

```
OS9:ezgen /d0/os9boot
```

using the usual R-command rules.

s — Save current module to a disk file. The file will be stored in the current execution directory. The name of the file will be the same as the module name.

s path — Save current module to a specified disk file.

t — Move module pointer to beginning of file.

u path — Update the current module using the module(s) in the file 'path'. The current module is deleted, and the contents of 'path' are inserted in its place.

v — Correct CRC & header parity of current module. This command is not allowed on data blocks.

x n — Extend current module by 'n' bytes. Bytes are added at the end of the module, and the CRC is corrected automatically.

y — Recalculate the size and offset of each module in the file. Use this command after modifying special data such as the module size in an 089 module header.

Notes:

* EZGen is used to modify a file containing one or more 089 modules. EZGen commands will add, delete, replace, patch, verify, or store the modules contained in the specified file.

* Two special file names are recognized by EZGen. If the name of the file is 089Boot, EZGen will automatically update the boot information in the ID sector of the device where the file is stored. Files named AltBoot cause an automatic update of the XT-ROM alternative boot information, which is stored in the last five bytes of the ID sector.

* A blank input line or end-of file has the same effect as the "Q" command.

* All EZGen commands operate on a copy of the input file. This work file is copied to the output file when either a blank line or the "Q" command is entered. The output file will have the same name as the original input file.

* Only the first character of an EZGen command is used by EZGen, so each command can be written as an entire word if desired. For example, you can use "rename" instead of "r" if you like.

* The EZGen utility normally aborts when an error occurs. If a command is prefixed with "-", then EZGen will not abort even if that command generates an error. The -N option can also be used to disable all error aborts.

```
EZGen Version 1.08
Copyright 1990 Burke & Burke

*** Link to old module
link d0
*** Change name
rename f0
*** All done.
q

OS9:
```

* To create a disk file containing a copy of the /h0 bootfile's "CC3GO" module:

```
OS9:ezgen /h0/os9boot

EZGen Version 1.08
Copyright 1990 Burke & Burke

*** Link to old module
l cc3go
*** Save as /h0/cmds/cc3go
s /h0/cmds/cc3go
*** All done.
q

OS9:
```

* To install a hard disk driver and device descriptor in the alternative boot file on drive /d0, replacing the existing /dd descriptor with a new version:

```
OS9:ezgen /d0/altboot

EZGen Version 1.08
Copyright 1990 Burke & Burke

*** Install new (hard disk) /dd
l dd
u /d0/modules/dd.dd
*** Link to RBF module -- put driver after it
l RBF
*** Move to next module, since we insert "before"
n
*** Read in the descriptors and driver
i /d0/modules/h0.dd
i /d0/modules/bbfhdisk.dr
*** All done.
q

OS9:
```

* To change the step rate of floppy drive /d2 to 6ms:

```
OS9:ezgen /d0/os9boot

EZGen Version 1.08
```

```
Copyright 1990 Burke & Burke

*** Link to device descriptor
l d2
*** Change step code to 3 (6 ms)
c 0014 3
*** Correct CRC
v
*** All done.
q

OS9:
```

* An example of the "-" prefix. In this example, an illegal module offset is specified in the "C" command.

```
OS9:ezgen foo
EZGen Version 1.08
Copyright 1990 Burke & Burke

*** Show module name
.
foo size=123
*** Change at illegal offset for example
-c 140 0 3
?(EZGen) illegal offset
*** Try again without the "-"
c 140 0 3
?(EZGen illegal offset

OS9:
```

* Another good use of EZGen is to add or update a module that has been MERGEd with other modules (e.g. Shell). In this case, the output file need not be contiguous, so you can speed up EZGen by using the '-f' option.

For example, to add the PARK command to your OS9 shell:

```
OS9:ezgen -f /d0/cmds/shell
link shell
next
insert /d0/cmds/park
quit
OS9:
```

TagTrack

Syntax: tagtrack [opts] source_path track_number

Function: Creates a file containing clusters on a particular track, and marks these clusters as "in-use".

Parameters:

opts	Any combination of the following options may be used to modify the operation of BootPort: -? Display options summary message. -e Mark entire track as in use. If this option is not specified, only those clusters that correspond to the 1st 18 sectors of the track are marked.
source_path	The name of the RBF device that contains the track that you wish to "tag".
track_number	The (decimal) track number to be "tagged".

Notes:

* TagTrack creates a file called "tag000" in the root directory of the source device. The segment list for this file contains exactly one entry, which includes all "tagged" clusters on the specified track.

* TagTrack will display a message indicating that a cluster is allocated to another file, but that cluster will still be allocated to "tag000". The result is a duplicate cluster assignment, which will be detected by the DCHECK utility. This allows TagTrack to be used to detect files that are resident on a particular track, as follows:

```
OS9:tagtrack /d0 34      (look for files on track 34)
.
.
OS9:dcheck -bp /d0
.
.
OS9:zap /d0/tag000       (undo the "tag" -- see ZAP)
OS9:
```

The output of the DCHECK command will include the names of all files that have clusters in common with "tag000".

* TagTrack can be used to delete the OS9 kernel from a hard disk or floppy disk by the sequence:

```
OS9:tagtrack /d0 34      (assumes floppy disk)
```

Zap

Syntax: zap [opts] path_name [path_name . . .]

Function: Removes directory entry and deallocates file descriptor sector for each specified file.

Parameters:

opts	Any combination of the following options may be used to modify the operation of BootPort: -? Display options summary message.
path_name	The name of the file that you wish to "zap". Any number of pathnames may be specified.

Notes:

* Zap does not deallocate the sectors that were specified in the segment list of the file. It simply releases the d directory entry and the file descriptor sector.

* One use of Zap is to mark bad sectors in the allocation bit map, so that OS9 will not attempt to use those sectors. This is done by creating a file that includes all of the bad sectors, and then Zap-ing the file.

* Another use of Zap is to deny OS9 access to groups of sectors that are used by other systems (e.g. the boot track). In this context, Zap and TagTrack (see TagTrack) perform opposite functions.

Example:

* To clean a file containing a bad sector from your disk, assuming the file is named "badfile":

```
OS9:zap /d0/badfile
OS9:
```

```
.
.
OS9:del /d0/tag000
OS9:
```

Use this sequence only if you are SURE that the data on track 34 is the OS9 kernel.

* Note that since TagTrack may create duplicate cluster assignments, it corrupts the file system in a predictable manner. The effect of TagTrack can be reversed by using the command:

```
OS9:zap /d0/tag000
```

* When using the EZGen utility to modify kernel data stored in tag000, be sure to specify the EZGen "-k" option.