esxDOS

User Manual

version 0.9.0 · November 2026

authors: SCjoe, lordcoxis

Contents

1 Introduction 3
2 Installation and card preparation 5
3 Drives, paths and file names 8
4 Working from BASIC 10
5 Tape, disk and snapshots 14
6 NMI menu and file browser 16
7 Configuration 17
8 Dot commands 18
9 Troubleshooting 20

Appendices

A NMI menu keys 21
B esxDOS dot commands 22
C Unofficial commands 25
D Configuration parameters 26
E Error codes 27
F Sources and links 28

Introduction

esxDOS is the ultimate firmware for the DivIDE, DivMMC and MB03+ interfaces that turns a classic ZX Spectrum, or a Timex Computer 2068, into a computer with a disk on an SD card, a CF card or an IDE hard drive.

What esxDOS does

esxDOS reads and writes FAT16 and FAT32 file systems and adds file-handling commands to ZX BASIC. It emulates a tape recorder with **.TAP** files and a Betadisk (TR-DOS) floppy drive with **.TRD** and **.SCL** images, and it loads **.SNA** and **.Z80** snapshots. The NMI button opens a file browser from which you start a program with a single key press.

Further functions come from dot commands: small programs in the `/SYS/BIN` and `/BIN` directories that you run from BASIC by typing their name with a leading dot, for example `.ls`. For programmers the system offers a POSIX-based interface called with the RST #08 instruction.

Supported hardware

The system runs on DivIDE and DivMMC interfaces (CF or SD card) for the ZX Spectrum 48K, 128K, +2, +2A/+2B, +3 and on compatible computers such as the Sizif-512, Omni128HQ, ZX-Uno, ZXDOS+ and eLeMeNt ZX. From version 0.9.0 it also runs on the Timex Computer 2068 (TC2068). For the firmware variants, see Chapter 2.

The latest version

If you have an older version, upgrade. Version 0.9.0 reads, creates and displays long file names (LFN) and adds BASIC commands: `MOVE` renames and moves files, `COPY ... TO` copies them, `NEW "dir"` creates a directory, `FORMAT` formats a drive, and `OPEN #`, `CLOSE #` and `POINT #` use files as BASIC streams. `LOAD`, `SAVE` and `VERIFY` handle arrays (`DATA`), file names can be any string expression, and `CAT`, `ERASE`, `MOVE` and `COPY` accept wildcards (for example `*.bin`). Most dot commands accept drive names and long names in quotes (`"hd1:/My Games"`), `.cp`, `.mv`, `.rm` and `.ls` also wildcards; the system commands are in `/SYS/BIN`, the others in `/BIN`. The firmware checks the version of the system files, the TC2068 is supported and many bugs are fixed.

The Timex Computer 2068

Thanks to Hugo Pinto, esxDOS runs on the Timex Computer 2068 with a DivIDE or DivMMC. Flash the TC2068 firmware for your interface and copy the same `/SYS` and `/BIN` directories as on a Spectrum; the TC2068 firmware also needs `/SYS/TC2068.KO`, which it loads before `ESXDOS.SYS` and from the same drive, and starts without the esxDOS commands if it is missing or of another version.

The commands work as on the Spectrum; only the Z80DMA settings do not apply on the TC2068. Hugo Pinto's TC2068 code also made `VERIFY` from a TAP file compare every byte, as with a real tape; the Spectrum firmware now uses the same code.

How to read this manual

Chapters 2 to 9 describe working with the system. All reference lists (keys, commands, configuration, error codes and links) are in Appendices A to F at the end.

In syntax descriptions, square brackets `[ ]` mark an optional part and a vertical bar `|` separates alternatives. Hexadecimal numbers are written with a `#` sign (for example `#2000`), binary numbers with a `%` sign.

Installation and card preparation

esxDOS consists of two parts: the firmware in the interface EEPROM and the system files on the card. Both parts must be the same version; otherwise esxDOS starts without its commands (see the warning below).

BEFORE YOU START

Prepare an SD card (a CF card or a disk for DivIDE), a card reader for your PC and the esxDOS 0.9.0 archive from esxdos.org. Back up the data on the card; formatting erases it.

1 Format the card

Format cards up to 2 GB as FAT16 and larger cards as FAT32. Use a single primary partition; esxDOS does not support extended partitions. On Windows, SD Card Formatter is a good tool; on Linux, use GParted. Larger allocation units (clusters, up to 32 KB) make the system respond faster.

2 Copy the system directories

Copy the `SYS`, `BIN` and `TMP` directories from the archive to the root of the card. On a freshly formatted card, copy them first so that the system finds them faster. Add other files (games, programs) afterwards.

3 Flash the firmware

Load the flasher tape for your interface and computer from the FLASHERS directory of the archive (see Firmware variants below): copy it to the card, attach it with `.tapein` and type `LOAD ""` (or load it from tape), then follow the instructions on the screen. The ROMS directory has the same firmware as plain files, for other flashing tools. The MB03+ and the eLeMeNt ZX are updated with the manufacturer's update package. When upgrading from a version later than 0.5, replace the system files on the card only after flashing the firmware.

4 Start the computer

Insert the card and switch the computer on. On the black start-up screen esxDOS lists the devices it found, their names (for example `hd0`) and the system files it is loading. Then the usual ROM message appears.

Firmware variants

esxDOS 0.9.0 has a firmware file for each interface, the DivIDE (IDE: CF card or disk) and the DivMMC (SD card), in one version for the ZX Spectrum and compatibles and one for the Timex Computer 2068. The firmware for the MB03+ and the eLeMeNt ZX comes with the eLeMeNt update package. Flash the file that matches your interface and computer. All firmware files use the same `/SYS` and `/BIN` files; the TC2068 also needs `TC2068.KO`.

Interface and computer	Firmware file
DivIDE (IDE), ZX Spectrum and compatibles	`esxdos-0.9.0-divide.bin`; with a flasher in `zx-ide-0.9.0.tap`
DivMMC (SD), ZX Spectrum and compatibles	`esxdos-0.9.0-divmmc.bin`; with a flasher in `zx-mmc-0.9.0.tap`

Interface and computer	Firmware file
MB03+ and eLeMeNt ZX	The esxDOS firmware for the eLeMeNt ZX computer and the MB03+ interface comes with the official eLeMeNt update package: https://mb03.elementhw.com/download
DivIDE (IDE), Timex Computer 2068	`esxdos-0.9.0-tc2068-divide.bin`; with a flasher in `tc2068-ide-0.9.0.tap`
DivMMC (SD), Timex Computer 2068	`esxdos-0.9.0-tc2068-divmmc.bin`; with a flasher in `tc2068-mmc-0.9.0.tap`

System directory structure

esxDOS looks for the system directories on the system drive `$`. At start-up this is the first drive with an `ESXDOS.SYS` of the firmware's own version (on the TC2068, with `TC2068.KO` on the same drive), or the first drive if none has one. If it is not the first drive, it also becomes the current drive, and the start-up screen shows its name after `ESXDOS.SYS`. A card can therefore keep the system files of an older esxDOS on `hd0`, where older firmware always looks, and those of 0.9.0 on another partition.

Location	Contents
`/SYS/ESXDOS.SYS`	System kernel
`/SYS/NMI.SYS`	NMI menu (file browser)
`/SYS/NMI_HELP.SCR`	NMI menu help screen (H key)
`/SYS/RTC.SYS`	Real-time clock driver for the RTC chip you have; not in the archive
`/SYS/BETADISK.SYS`	TR-DOS (Betadisk) emulator
`/SYS/*.KO` – BASIC commands	`BLOAD.KO`, `BSAVE.KO`, `BCAT.KO`, `BDIR.KO`, `BCOPY.KO`, `BFORMAT.KO`, `BOPEN.KO`, `BSTREAM.KO` – loaded only when a command needs them
`/SYS/*.KO` – other modules	`TAPE.KO`, `VDISK.KO`, `AUTOLOAD.KO`, `ERRMSG.KO`, `TRDOS54T.KO`
`/SYS/TC2068.KO`	Timex Computer 2068 support (TC2068 only)
`/SYS/CONFIG`	Configuration files
`/SYS/BIN`	System dot commands
`/BIN`	Other dot commands
`/TMP`	Temporary files

> **WARNING**
>
> Use the system files from the same archive as the firmware in the EEPROM. From version 0.9.0 the firmware checks the version of `ESXDOS.SYS` , `BETADISK.SYS` and, on the TC2068, `TC2068.KO`: a file of another version is not loaded and the start-up screen shows [ERROR] after its name (for `TC2068.KO`, after `ESXDOS.SYS`). `ESXDOS.SYS` and `TC2068.KO` are looked for on every drive, so for them [ERROR] means that no drive has them in the firmware's version. Without `ESXDOS.SYS`, BASIC starts without the esxDOS commands and without the NMI menu; without `BETADISK.SYS`, only TR-DOS is missing. `NMI.SYS` is not checked: an `NMI.SYS` from an older version crashes when NMI is pressed. Older firmware does not check, and a mismatch locks up the system. You can set the Hidden and Read-only attributes on the system files, for example `.chmod +h /sys/esxdos.sys`.

Reinitialisation

After changing the card, press the RESET button while holding down the **SPACE** key. esxDOS searches for devices again and reloads the system files. If the card is still not found, switch the computer off and on.

The card in an emulator

The LnxSpectrum, Fuse, ZEsarUX, eLeMeNtemu and Retro Virtual Machine emulators can emulate DivMMC or DivIDE with esxDOS on top of a card image. Fuse uses **HDF** images, Retro Virtual Machine **IMG** images. LnxSpectrum, eLeMeNtemu and ZEsarUX work with both HDF images and raw (RAW) **IMG** images. These three emulators can also use a folder on the PC disk instead of an image, so you do not have to copy files into an image; in ZEsarUX this is enabled by the esxDOS handler option. On Windows you can edit the contents of an HDF image with the HDF Monkey tool.

Drives, paths and file names

Every file is identified by its drive, path and name. If you omit the drive, the current drive applies; if you omit the path, the current directory applies.

Drive names

A drive is named after the device type and the partition number: **hd0** is the first partition found, **hd1** the second. esxDOS prints the exact names at a cold start, and the `.df` command lists them too, with their size and free space (`.df -h` in K, M or G). Two shortcuts always work: `*` is the current drive and `$` is the system drive esxDOS booted from.

> **DRIVES ARE NOT SLOTS**
> Do not confuse drives (partitions) with card slots. If one medium (SD card, CF card or disk) contains several FAT partitions, esxDOS presents them as separate drives. If one medium has four partitions, they become drives `hd0` to `hd3`, and the partitions on the second card or disk follow from `hd4`.

Paths

Directories are separated by a slash `/` as in Unix. A path starting with a slash leads from the root of the drive (`/games/manic`), a path without a slash leads from the current directory (`manic`). Two dots `..` mean the parent directory.

BASIC commands (Chapter 4) and the `.cd`, `.chmod`, `.cp`, `.divideo`, `.file`, `.ls`, `.lstap`, `.mkdir`, `.mktrd`, `.mv`, `.rm`, `.sercp`, `.snapload`, `.tapein`, `.tapeout`, `.trkplay` and `.vdisk` dot commands accept a drive name with a colon before the path. A colon outside quotes ends a BASIC statement, so a path with a drive name goes in quotes, for example `.ls "hd1:/demos"`; `"hd1:"` on its own means the current directory of that drive. Names containing spaces go in quotes too; for example, `.mv "hd0:/Games/Midnight Resistance.tap" "hd1:"` moves the file to the second drive. Between drives `.mv` copies the file and then deletes the original; a directory can only be moved within its drive.

File names

esxDOS reads, creates and displays long file names (LFN) of up to 65 characters, so you see files from a PC under the same names as on the computer. Long names are used when `ESXDOS.CFG` contains `LFN=1`, as the supplied file does (Chapter 7); without it esxDOS works with the short 8.3 names only. A file whose long name is longer than 65 characters is shown under its short name, and such a name cannot be created ("Path too LONG"). A path may have at most 126 characters. Letter case does not matter.

Wildcards

The `CAT`, `ERASE`, `MOVE` and `COPY` commands and the `.cp`, `.mv`, `.rm` and `.ls` dot commands accept wildcards in the last part of a name: `*` stands for any run of characters, `?` for any one character, upper or lower case alike; for example `*.bin` or `/games/a?.z80`. A name starting with a dot matches only a pattern that starts with one. When a pattern selects files to copy, move or delete, directories and hidden files are left out; `.cp` and `.mv` also leave out system files, and `.rm -r` removes matching directories too.

File extensions

The type of a BASIC file (program, array, code) is determined by the +3DOS header, not by the extension, so extensions are only a convention and you can name your files as you like. The table lists the extensions this manual suggests; `.BAS` and `.BIN` follow the convention of the +3 manual. Some tools may

use others: the LnxCopy copier, for example, adds `.COD`, when converting from tape or a TAP file to the card, and saves headerless records as `FILE_000.COD` to `FILE_999.COD`.

Extension	Contents
`.BAS`	BASIC program.
`.BIN or .COD`	Code block (CODE), as on the +3; also data saved with `SAVE ... BIN`, without a header.
`.SCR`	Screen (SCREEN$), 6912 bytes; 6144 bytes without attributes.
`.NUM`	Number array.
`.STR`	String (character) array.
`.ROM`	Alternative ROM image, 16384 bytes, headerless.
`.TAP`	Tape emulator file containing a sequence of records.

DIRECTORY LIMIT

The NMI browser shows at most 704 files in one directory. Split large collections into subdirectories, for example by initial letter.

Working from BASIC

esxDOS extends ZX BASIC commands with a drive name and a path, and adds the `MOVE`, `COPY`, `NEW "dir"` *and* `FORMAT` *commands and file streams. It stores BASIC files on FAT with a +3DOS-compatible header, so the file type, start line and code address are preserved. The drive goes in the name, before the path:* `LOAD *"hd1:/games/jet.bas"`, `CAT "hd1:"` (`"hd1:"` *alone is that drive's current directory). It is checked only when the line runs, so a program can name a drive that is not mounted yet.*

DEPRECATED · DRIVE BEFORE THE NAME

Before 0.9.0, BASIC commands took the drive in front of the name: `LOAD hd1"game"`, `CAT hd1`, `GOTO hd1"games"`. This form still works in 0.9.0 but is deprecated and will be removed in version 1.0. Write the drive inside the name instead: `LOAD *"hd1:game"`, `CAT "hd1:"`, `GOTO "hd1:games"`.

Changing drive and directory: GOTO

SYNTAX

```
GOTO ["[drive:]path"]
GOTO variable$
```

`GOTO` without parameters prints the current drive and path. With parameters it changes the drive, the directory or both. The path can be passed in a string variable.

EXAMPLE · ZX BASIC

```
GOTO "hd0:games"
LET n$="/demos": GOTO n$
GOTO "hd1:"+n$
```

Listing a directory: CAT

SYNTAX

```
CAT [*] ["[drive:]name"]
```

`CAT` without a drive gives a fast listing without header information. With a drive, `CAT *` for the current one or `CAT "hd1:"` for another, it prints a detailed list with the header information, the number of items and the free space. A name lists another directory or, with wildcards in its last part, only the matching names (files and directories): `CAT "*.TAP"`, `CAT *"GAMES/A*.TAP"`. The name can be any string expression, for example `CAT a$`. Each file's size is shown in K or M (a directory as `<DIR>`); for dates and exact sizes, use the `.ls -l` dot command.

Loading a file: LOAD, VERIFY and MERGE

SYNTAX

```
LOAD *"[drive:]path/file" [CODE [address[,length]] | SCREEN$ | DATA array()]
LOAD *[;]name$
VERIFY *"[drive:]path/file" [CODE [address[,length]] | SCREEN$ | DATA array()]
MERGE *"[drive:]path/file"
```

The asterisk is mandatory with LOAD, otherwise the data are read from tape; another drive goes in the name, as in `LOAD *"hd1:/games/jet.bas"`. The name can be any string expression, for example `LOAD *a$` or `LOAD *"hd1:games/"+n$`; the older form with a semicolon, `LOAD *;n$`, still works. VERIFY compares a file with memory; MERGE adds the lines of a program to the one in memory. `LOAD ... DATA` creates the array under the name given in the command, replacing an array of that name; a number array and a character array cannot be mixed ("Wrong file TYPE").

EXAMPLE · ZX BASIC

```
LOAD *"/progs/editor.bas"
LOAD *"picture.scr" SCREEN$
LOAD *"routine" CODE 40000
LET n$="game": LOAD *;n$
LOAD *"scores.num" DATA s()
LOAD *"names.str" DATA n$()
VERIFY *"routine" CODE 40000
```

Saving a file: SAVE

SYNTAX

```
SAVE *"[drive:]path/file" [LINE n] | [CODE addr,length] | [BIN addr,length] |
[SCREEN$] | [DATA array()]
```

The CODE variant saves a block with a +3DOS header. The BIN variant saves only the raw data without a header, so a PC can read the file too. The DATA variant saves a number array (for example `s()`) or a character array (for example `n$()`); `LOAD ... DATA` loads it back under the given array name. If the file exists, SAVE asks "File exists, overwrite? (Y/N)". To save a screen in the standard SCR format, use SAVE `*"picture.scr" BIN 16384,6912`.

EXAMPLE · ZX BASIC

```
SAVE *"menu" LINE 10
SAVE *"picture" SCREEN$
SAVE *"/dat/table.bin" BIN 40000,2500
SAVE *"scores.num" DATA s()
SAVE *"names.str" DATA n$()
```

STRINGS

DATA saves only arrays, as on tape. To keep an ordinary string, copy it into a character array first (`DIM t$(LEN a$): LET t$=a$`) or write it to a file stream (see below).

Deleting a file: ERASE

SYNTAX

```
ERASE "[drive:]path/file"
```

ERASE deletes a file or an empty directory. With a mask in the last part of the name (for example `*.bin`) it deletes all matching files, but not directories or hidden files, stopping at the first error. The name can be any string expression. To delete a directory with its contents, use the `.rm` dot command.

Renaming and moving: MOVE

SYNTAX

```
MOVE "[drive:]old" TO "new"
```

The MOVE command renames a file or a directory, or moves it to another directory on the same drive. If the new name is an existing directory, the file moves into it; if it is an existing file, MOVE stops with "File already EXISTS". With a mask, every matching file moves into the directory given as the new name. Both names can be string expressions; a drive in the new name must be the old one's. To move files to another drive, use the .mv dot command.

EXAMPLE · ZX BASIC

```
MOVE "game.tap" TO "/games/game.tap"
MOVE "*.scr" TO "/pictures"
```

Copying a file: COPY

SYNTAX

```
COPY "[drive:]source" TO "target"
```

COPY copies a file, on the same drive or to another one (`COPY "game.tap" TO "hd1:/backup"`). The target can be a new name, an existing file, which is overwritten, or a directory, which receives a copy under the same name. With a mask in the source every matching file is copied into the target directory, stopping at the first error. Both names can be string expressions; without a drive the new name is on the current drive, as with the .cp dot command.

EXAMPLE · ZX BASIC

```
COPY "game.tap" TO "backup.tap"
COPY *"*.tap" TO "/backup"
```

Creating a directory: NEW

SYNTAX

```
NEW "[drive:]directory"
```

NEW followed by a name creates a directory, as the .mkdir dot command does: `NEW "games"`, `NEW "hd1:/demos"`, `NEW a$`. NEW on its own still clears the program.

Formatting: FORMAT

SYNTAX

```
FORMAT device ["label"]
```

FORMAT creates a new, empty FAT file system. Name the device as the start-up screen lists it: a partition (hda1, sdb2) or a whole device (hda, sdb), which is then formatted without partitions, like a floppy disk. Unmount its drives first with .umount, otherwise FORMAT stops with "Device is BUSY"; for example, if the start-up screen showed hd1 on hda2, type `.umount hd1` and then `FORMAT hda2 "GAMES"`. Devices smaller than 512 MB get FAT16, larger ones FAT32, up to 128 GB; the minimum is 4 MB. The label can be any string expression; its first 11 characters are used, in upper case. FORMAT asks for confirmation (Y/N) and then prints a dot every 256 sectors.

WARNING

FORMAT erases everything on the partition or device. Formatting a whole device also removes its partitions.

Files as streams: OPEN #, CLOSE # and POINT

SYNTAX

```
OPEN #stream;"[drive:]path/file" [IN | OUT | RND]
CLOSE #stream | CLOSE #* | CLEAR #
POINT #stream,position
(POINT #stream)   (LEN #stream)
```

A file can be attached to a BASIC stream and used with `PRINT #`, `INPUT #` and `INKEY$ #`, as on the Interface 1 or with Uni-Dos. Use streams 4 to 15. `IN` opens an existing file for reading, `OUT` creates a file or empties an existing one, `RND` opens a file for reading and writing and creates it if needed; without a mode the file is read if it exists and created otherwise. The name can be any string expression.

`POINT #` moves to a byte position counted from 0; position = the length adds to the end of the file. In an expression, `(POINT #s)` and `(LEN #s)` give the position and the length; the brackets are required. `INPUT #4; LINE a$` reads one line of text; at the end of the file `INPUT #` stops with "8 End of file", while `INKEY$ #` returns an empty string. `CLOSE #` writes the remaining data and closes the file, `CLOSE #*` closes all files, and `CLEAR #` also resets streams 0 to 15.

To append to an existing file, open it with `RND`, which keeps what is in it, and move to its end before writing: `OPEN #4;"/tmp/log.txt" RND: POINT #4,(LEN #4): PRINT #4;"third": CLOSE #4`. `OUT` would empty the file first.

EXAMPLE · ZX BASIC

```
10 OPEN #4;"/tmp/log.txt" OUT
20 PRINT #4;"first": PRINT #4;"second"
30 CLOSE #4
40 OPEN #4;"/tmp/log.txt" IN
50 INPUT #4; LINE a$: INPUT #4; LINE b$
60 PRINT a$;"/";b$;" length ";(LEN #4)
70 CLOSE #4
```

+3DOS headers

A file with a header carries its type (program, array, code), length and address or start line, like a tape record. Headerless files (pictures from a PC, data) are loaded by giving the address (or `SCREEN$`); the length is the file's own unless one is given. The `.file` command identifies the file type.

WARNING

`LOAD ... CODE` writes only to RAM (`#4000–#FFFF`), as a tape load does: bytes that would go below `#4000` are skipped. Before 0.9.0, loading into `#2000–#3FFF` overwrote esxDOS and crashed the system. `SAVE ... CODE` and `VERIFY ... CODE` of `#0000–#3FFF` still see esxDOS's own memory there instead of the Spectrum ROM, so they cannot save or check the ROM.

SUGGESTED EXTENSIONS

Extensions are only a convention: esxDOS reads the type of a file from its header. This manual suggests program `.BAS`, code `.BIN` (as on the +3), screen `.SCR`, number array `.NUM` and string array `.STR`.

Tape, disk and snapshots

esxDOS emulates a tape recorder and a Betadisk floppy drive, so you can run tape and disk games from the card without changes.

Tape emulation

The `.tapein file.tap` command attaches a TAP file to the tape input. A following `LOAD ""` reads blocks from the file instead of the tape, including further parts that a program loads through the ROM loading routine; programs with their own (turbo) loaders cannot load from a TAP file. The `-a` switch performs a fast reset and an automatic `LOAD ""`. The `-p` switch toggles the pause after a screen is loaded, `-r` rewinds the file to the beginning and `-c` detaches it.

The `.tapeout file.tap` command redirects `SAVE` without a drive to a TAP file, where blocks are appended at the end. The `-o` switch overwrites the file, `-c` detaches it. `.lstap -i` and `.lstap -o` list the contents of the attached files.

EXAMPLE · ZX BASIC

```
.tapein /games/manic.tap
LOAD ""
.tapeout backup.tap
SAVE "data" CODE 40000,1000
.tapeout -c
```

PARITY CHECK

The parity check when reading TAP files is off by default. Turn it on with the `ParityCheck=1` parameter in `ESXDOS.CFG`.

Snapshots

Load SNA and Z80 snapshots with the `.snapload file` command or from the NMI browser. The `-l` switch runs a snapshot in 48K mode with paging locked, `-u` ignores the lock stored in the snapshot. If you hold CAPS SHIFT while loading, a menu appears for choosing the RAM page, ROM, video memory and paging lock manually.

To save a snapshot of a running program, press **S** in the NMI menu; the files are named `snap0000.sna`, `snap0001.sna` and so on. 128K snapshots often work only on the original 128K model, not on the +2A, +2B and +3.

TR-DOS (Betadisk) emulation

Attach a disk image with the `.vdisk` unit file command, where units 0 to 3 correspond to drives A to D. The file can have a drive name and a long name, in quotes, for example `.vdisk 0 "hd1:/trd/game.trd"`. The unit number alone detaches the image. TRD and SCL images can also be attached from the NMI browser with the A to D keys. TR-DOS is then operated as on a real Betadisk.

`.mktrd file.trd [label]` creates an empty disk image (double-sided, 80 tracks); a drive name goes in quotes, for example `.mktrd "hd1:/trd/new.trd" GAMES`. The `.scl2trd` command converts an SCL image to TRD; with the `f` parameter it creates a full 640 KB image, without it a trimmed one. The emulator behaviour is set in the `TRDOS.CFG` file (Appendix D).

EXAMPLE · ZX BASIC

```
.vdisk 0 /trd/game.trd
RANDOMIZE USR 15616
.vdisk 0
```

TZX

esxDOS does not load TZX files yet. Convert them to TAP on a PC, unless they use non-standard loading routines.

NMI menu and file browser

The NMI button on the interface stops the running program at any time, saves its state and opens the file browser. Press SPACE to return to the program as if nothing had happened.

The NMI menu is a separate program, `/SYS/NMI.SYS`, written by ub880d, which esxDOS loads at start-up into the interface memory. It recognises the most common formats and starts a file with the ENTER key: it attaches and loads TAP, restores SNA and Z80, displays SCR, attaches TRD or SCL and starts TR-DOS, and loads and runs a BASIC program.



NMI menu help screen (H key)

The full list of keys is in Appendix A. The controls differ slightly between `NMI.SYS` versions; you can always call up the help with the **H** key.

Alternative browsers

Instead of the standard NMI menu you can use an extended browser by bob_fossil (thefossilrecord.co.uk website). It views texts and pictures, plays music and can copy and delete files. You can also call up the original menu from it with CAPS SHIFT+N. Browsers made for esxDOS 0.8.x do not run on 0.9.0, which loads `NMI.SYS` at #2000; use a version built for 0.9.0.

Configuration

The behaviour of esxDOS is controlled by text files in the /SYS/CONFIG directory. The defaults suit most set-ups; you rarely need to change them.

The `ESXDOS.CFG` file configures the kernel: automatic start of a program after boot, the colour for automatic LOAD, long file names, Z80DMA controller timings and the tape parity check. The `TRDOS.CFG` file configures TR-DOS emulation. Each line has the form `Parameter=value`; lines starting with # are comments. All parameters are in Appendix D.

You can edit the configuration directly on the Spectrum with the `.te` text editor, for example `.te /sys/config/esxdos.cfg`. The ENTER key writes a change to the file immediately; BREAK quits the editor. esxDOS reads the configuration only at a cold start (power on, or RESET with SPACE held), so a change takes effect then.

Automatic start

The `AutoBoot` parameter determines when esxDOS runs the `/SYS/AUTOBOOT.BAS` program: 0 never, 1 only after a cold start, 2 only after a warm start (RESET), 3 always. Holding CAPS SHIFT during start-up does the opposite: it skips `AUTOBOOT.BAS`, or runs it when `AutoBoot` would not. Save into `AUTOBOOT.BAS`, for example, a menu that sets the directory and offers games.

EXAMPLE AUTOBOOT.BAS · ZX BASIC

```
10 GOTO "hd0:games"
20 .ls
SAVE *"/sys/autoboot.bas" LINE 10
```

Other configuration files

Some dot commands have their own configuration in the same directory. For example, `.sercp` reads the default interface and transfer speed from `/SYS/CONFIG/SERCP.CFG`.

Dot commands

A dot command is a small program in the `/SYS/BIN` *or* `/BIN` *directory. You run it by typing its name with a leading dot; esxDOS loads it into the interface memory, runs it and returns to BASIC. The system commands (the core ones, such as* `.ls`, `.cp` *or* `.tapein`*) are in* `/SYS/BIN` *and all others, including the utilities that come with esxDOS, in* `/BIN`*; esxDOS looks in* `/SYS/BIN` *first, so when both have a command of the same name, the one in* `/SYS/BIN` *runs. A command name can have up to 26 characters.*

A dot command can be entered as a direct command or as part of a program line, and it can be combined with other statements after a colon. Parameters are separated by spaces. Most commands that need a parameter print a short usage with their switches when run without one.

EXAMPLE · ZX BASIC

```
.ls
.cd /games
10 .cp game.tap backup.tap: PRINT "OK"
```

The set of commands differs between distributions and versions. To see what you have, list `.ls /SYS/BIN` and `.ls /BIN`; a command's file name is also its name. To install a new command, copy its file into `/BIN` without an extension; `/SYS/BIN` holds esxDOS's system commands.

Command groups

The esxDOS distribution commands cover file handling (`.ls`, `.cd`, `.cp`, `.mv`, `.rm`, `.mkdir`, `.chmod`, `.tar`, `.touch`, `.fm`), viewing (`.more`, `.hexdump`, `.hexview`, `.file`, `.strings`, `.sview`), editing (`.te`, `.vi`, `.editor`, `.fontedit`), tape and disks (`.tapein`, `.tapeout`, `.lstap`, `.vdisk`, `.mktrd`, `.scl2trd`, `.snapload`), music, sound and video playback (`.trkplay`, `.playwav`, `.divideo` and others), data transfer (`.sercp`), system services (`.drives`, `.partinfo`, `.date`, `.cal`, `.dumpmem`, `.dfwload`) and programming tools (`.dbasic`).

Other useful commands

The `.launcher` command creates a link to a TAP or BAS file, which you then start by name. `.launcher add manic /games/manic.tap` creates a link, `.launcher list` lists existing links, `upd` and `del` change or delete them.

The `.sercp` command transfers files over a serial line to and from a PC or another Spectrum: through the AY serial port of the 128K, +2, +2A and +3 (`-a`, default), the eLeMeNt ZX or MB03+ USB port (`-e`), the ZX Next ESP serial port (`-x`) or the SIF interface (`-s`), at 38400 Bd (`-n`, default) or 115200 Bd (`-t`). `.sercp file` sends a file (drive names and long names in quotes), `-r` receives into the current or the given directory; the defaults come from `/SYS/CONFIG/SERCP.CFG`. The PC side is the sercp program from vym.cz.

EXAMPLE · ZX BASIC

```
.sercp -r /games
.sercp /dat/results.txt
```

Commands for other hardware

Some dot commands are written for particular interfaces, such as the MB03+, or for computers such as the ZX-Uno, the ZX Spectrum Next and the eLeMeNt ZX. They are not part of the standard esxDOS distribution and this manual does not describe them.

Troubleshooting

Most problems are caused by the version of the system files, the card format or the file name. Start with the simplest checks.

esxDOS does not appear at power-on

Check that the interface is enabled (switch or jumper) and that the card is formatted as FAT16 or FAT32 with a single primary partition. Clean the interface connector and plug it in again. Try RESET with the SPACE key held down.

Some TVs and monitors supply the computer with a small current through the digital video cable even when it is switched off. The card then stays locked. Disconnect the video cable and switch the computer off completely.

The system does not start after an upgrade

No drive has an `ESXDOS.SYS` of the same version as the firmware in the EEPROM. The start-up screen then shows [ERROR] after `ESXDOS.SYS` and BASIC starts without the esxDOS commands; older firmware locks up. Flash the firmware and copy the system files from the same archive.

[ERROR] appears at start-up

A file that esxDOS loads at start-up is missing from the `/SYS` directory, most often `NMI.SYS`, `RTC.SYS` or `BETADISK.SYS`, or is of another version (`ESXDOS.SYS`, `BETADISK.SYS`, `TC2068.KO`). Restore it from the archive. `ESXDOS.SYS` and `TC2068.KO` are looked for on every drive, so for them [ERROR] means that no drive has them in the firmware's version. `RTC.SYS` is not in the archive: it depends on the RTC chip you have, and without one the [ERROR] after `RTC.SYS` is normal. The system runs without `NMI.SYS`, `RTC.SYS` or `BETADISK.SYS`, only without that function. Error 26 "Loading .KO FAILED" when you use a BASIC command means that its module is missing from `/SYS`.

A file cannot be found or loaded

Check the exact file name with `.ls`. Check the current drive and directory with `GOTO` without parameters. Error 7 "Invalid FILENAME" means a character that is not allowed in the name.

A program crashes or misbehaves

Some programs do not work on machines with esxDOS because they use the interface memory area or ports. It may help to run them with paging locked (`.snapload -l`), to load them through `.tapein` instead of a snapshot, or to disable the interface temporarily.

> **IF IT STILL DOES NOT WORK**
>
> Post a description of the problem with the esxDOS version, interface type and computer model on the board.esxdos.org forum. The list of error codes is in Appendix E.

NMI menu keys

You can always call up the help with the H key.

Key	Function
Arrow up / down (7 / 6)	Move the cursor by one entry.
Arrow left / right (5 / 8)	Previous / next page of the list.
ENTER	Enter a directory; for a file, load and run it (TAP, SCR, SNA, Z80, TRD, SCL, BASIC); 0 does the same.
EDIT (1)	Parent directory.
V	View the screen from an SCR, SNA or CODE file with a header.
I	Attach a TAP file to the tape input (tapein).
O	Attach a TAP file to the tape output (tapeout).
A, B, C, D	Attach a TRD or SCL image to drive A to D.
S	Save a snapshot `snapxxxx.sna`; error 18 means the file exists, try again.
R	Warm reset.
SPACE	Leave the NMI menu and return to the interrupted program.
H	On-screen help.

esxDOS dot commands

The syntax shown is that of the 0.9.0 distribution, in which a drive name is written before the path with a colon; a path with a drive name goes in quotes, for example `"hd1:/games"`.

System commands (/SYS/BIN)

Command	Syntax	Description
.cd	.cd [drive:][path]	Changes the current directory and shows it; a drive name also makes that drive the current one.
.chmod	.chmod +|-rwxash [drive:]file...	Sets or clears attributes: w writable (-w makes a file read-only), a archive, s system, h hidden; r and x are accepted but have no effect; from 0.9.0 several files.
.cp	.cp [drive:]source... [drive:]target	Copies files; from 0.9.0 between drives, with long names and wildcards. Several sources or a wildcard need a directory as the target.
.date	.date	Prints the date and time from the real-time clock (RTC.SYS); it cannot set them.
.df	.df [-h]	Lists the drives with their volume name, file system, size and free space in K; -h in K, M or G. In BASIC's 32 columns always in K, M or G and without the file system.
.file	.file [drive:]file...	Identifies multiple Spectrum-related filetypes. From 0.9.0 several files at once, with drive names and long names.
.ls	.ls [-l] [-h] [-a] [-t] [-r] [[drive:]name...]	Lists directories, names in columns; -l one a line with attributes, size and date, -h with -l sizes in K, M or G, -a hidden files too, -t newest first, -r in reverse order; in version 0.9.0 with drive names (hd1:), long file names and wildcards.
.lstap	.lstap -i|-o|[drive:]file.tap	Lists the blocks of a TAP file, or of the attached input (-i) or output (-o) file.
.mkdir	.mkdir [drive:]directory...	Creates directories; from 0.9.0 several at once.
.more	.more [-a] file	Paged text display; -a without converting line endings.
.mount	.mount device	Mounts a partition or device as a drive, for example .mount hda2.
.mv	.mv [drive:]source... [drive:]target	Renames or moves files and directories; from 0.9.0 with wildcards, long names and between drives (a file is copied, then deleted; a directory only within its drive).
.rm	.rm [-rfvi] [drive:]name...	Removes files; -r directories and everything in them, -f no error for missing names and read-only files too, -v shows each one removed, -i asks first. Wildcards in the last part of a name; a drive's root is never removed.
.sercp	.sercp [-aexsnt] [drive:]file .sercp [-aexsnt] -r [[drive:]path] .sercp -h	Serial transfer to and from a PC or another Spectrum (v0.9, Pavel Vymětálek): -a the AY serial port of the 128K, +2, +2A and +3 (default), -e eLeMeNt ZX or MB03+, -x ZX Next, -s SIF interface; -n 38400 Bd (default), -t 115200 Bd. Defaults from /SYS/CONFIG/SERCP.CFG. From v0.9 with drive names and long names.
.snapload	.snapload [-l|-u] [drive:]file	Loads an SNA or Z80 snapshot; -l 48K mode, -u ignores the lock.
.strings	.strings [-f] [-n length] [-t d|x] file	Prints the readable strings in a file, optionally with their offsets.
.tapein	.tapein [-c|-p|-a|-r|-s pos|-i] [[drive:]file.tap]	Attaches a TAP file to the tape input; -p toggles the SCREEN$ pause; see Chapter 5.

Command	Syntax	Description
.tapeout	.tapeout [-c|-o|-i] [[drive:]file.tap]	Redirects SAVE to a TAP file.
.tar	.tar -c|-t|-x [-v] archive [files]	Creates, lists or extracts a TAR archive (UB880D).
.te	.te file	Text editor for configuration and text files (nagydani).
.trkplay	.trkplay [drive:]file	Plays ProTracker 3 (and Vortex Tracker II), Sound Tracker and SQ Tracker music on the AY, with a meter for each channel; SPACE stops it (players from gasman's trackerplayer).
.umount	.umount drive	Unmounts a drive, for example before FORMAT; refused while a file on it is open.
.vdisk	.vdisk unit [[drive:]file]	Attaches a TRD or SCL image to drive 0–3 (A–D); without a file it detaches the drive.

Other commands (/BIN)

Command	Syntax	Description
.128	.128	Starts 128K BASIC on 128K machines (fixed in version 0.8.8).
.cal	.cal	Monthly calendar from the real-time clock (UB880D).
.dbasic	.dbasic [action ...] [undo]	Toolkit for the BASIC program in memory: info, list, find, vars, delete, rem, reset, compress, renumber, undo (128K); without an action it lists them (namco).
.dfwload	.dfwload file	Loads other firmware into the interface RAM for testing without rewriting the EEPROM (ub880d).
.divideo	.divideo [drive:]file.dvo	Plays a DivIDEo video stream; DivIDE only. From 0.9.0 (v1.1) with drive names and long names.
.drives	.drives [-1|-2|-3]	Lists drives and devices; the switch selects the system function (Dr. Slump).
.dskprobe	.dskprobe	Sector viewer for devices and virtual disks.
.dumpmem	.dumpmem [-s<addr>] [-l<length>] [-f] file	Saves a memory area to a file; default address 32768, length 512, -f overwrites the file.
.editor	.editor file	Hex and ASCII file viewer and editor: changes bytes in place, finds text or bytes, saves a block to a file or tape (namco).
.fm	.fm	Two-panel file manager: copies, moves, erases and renames files and directories, also between drives; views pictures, texts and fonts, runs programs (namco).
.fontedit	.fontedit file	Font editor for 768-byte Spectrum fonts; .FNT is added to a name without an extension, a new file starts from the ROM font (namco, from ZX ArtWorks 128).
.gramon	.gramon	Graphics memory monitor for finding and viewing graphics.
.hexdump	.hexdump file	Prints a file in hexadecimal.
.hexview	.hexview file	Interactive hex viewer, paging with the arrow keys (Dr. Slump).
.launcher	.launcher add|upd name /path/file·del name·list	Manages links to TAP and BAS files, given with their full path (UB880D).
.mktrd	.mktrd [drive:]file.trd [label]	Creates an empty TR-DOS disk image (double-sided, 80 tracks); a label with spaces in quotes.
.partinfo	.partinfo	Lists the MBR partitions on the connected devices.
.playtfm	.playtfm file.tfc	Plays TurboSound FM music.
.playwav	.playwav file.wav	Plays WAV sound.
.scl2trd	.scl2trd [f] file.scl	Converts SCL to TRD; f creates a full image.

Command	Syntax	Description
.speakcz	.speakcz "TEXT"	Czech speech synthesis; upper-case letters only (port by Velesoft).
.sview	.sview [-b<bank>] [address] · .sview file [offset]	Shows memory or a file as graphics, to find sprites, tiles and fonts; saves the screen as an SCR file (namco, after SuperView, MicroHobby 1987).
.touch	.touch file	Sets the date and time of an existing file to the current ones from the real-time clock; it does not create a file (ub880d).
.tsplay	.tsplay file	Plays TurboSound music (two AY chips).
.vi	.vi [file] [32|42|64]	Text editor in 32, 42 or 64 columns, with find, undo and a : command line as in the Unix vi (namco).

Unofficial commands

Read the command help before use.

Command	Description	Author
`.load`	Loads a BASIC program from a text file.	nagydani
`.save`	Saves a BASIC program as text.	nagydani
`.merge`	Merges a BASIC program from a text file.	nagydani
`.open` / `.close`	Opens or closes a file as a BASIC stream for `PRINT #` and `INPUT #`; syntax `.open #stream file`.	nagydani
`.play`	Plays a melody written like the PLAY command on the +3.	A. Owen
`.kzx`	Shows the keyboard layout.	samista
`.testram`	48K and 128K memory test.	
`.zxzvm`	Z-machine player for text adventures.	bob_fossil
`.playay`	AY music player.	bob_fossil
`.browse`	Extended file browser.	bob_fossil
`.tap2mic`	Writes a TAP file to real tape through the MIC output.	rkd77
`.dzx7`	Decompresses ZX7-compressed files.	Alcoholics Anonymous
`.gsc`	Controls the General Sound card, plays MOD files.	Nihirash
`.playzxm`	Plays ZXM video (colour or black and white).	

Configuration parameters

esxDOS reads its settings from two text files in `/SYS/CONFIG`*. The value in the supplied file is shown in brackets; a parameter left out of the file is 0, except* `AutoLoadColour`*, which is 7.*

/SYS/CONFIG/ESXDOS.CFG

Parameter	Values	Meaning
`AutoBoot`	0–3 (0)	Runs `/SYS/AUTOBOOT.BAS`: 0 never, 1 after a cold start, 2 after a warm start, 3 always.
`AutoLoadColour`	attribute (7)	Screen attribute after an automatic LOAD: PAPER×8+INK (+64 BRIGHT, +128 FLASH); 7 is white ink on black paper.
`DMA`	0 | 1 (0)	Use of the Z80DMA controller, if the hardware has one (MB03+, eLeMeNt ZX); not on the TC2068.
`DMA_IO`	2–4 (4)	DMA timing for ports .
`DMA_RAM`	2–4 (4)	DMA timing for memory.
`ParityCheck`	0 | 1 (0)	Parity check of TAP blocks.
`LFN`	0 | 1 (1)	Long file names on every drive.

/SYS/CONFIG/TRDOS.CFG

Parameter	Values	Meaning
`LoadMode`	0 | 1 (1)	0 standard, 1 STEALTH: loads more programs but needs a stable bus.
`ReadDelay`	number (32)	Slows reading down to the speed of a real TR-DOS; 0 means no delay.
`WriteDelay`	number (0)	Slows writing down.
`AutoIM2`	0 | 1 (1)	Automatic detection of the interrupt mode in TR-DOS snapshots (Velesoft).
`SafeCallBAS`	0 | 1 (0)	Set to 1 if an IM2 loader ends with a red border.
`AlwaysBoot`	0 | 1 (1)	If the disk has no boot file, runs `/SYS/TRDBOOT.BIN`.
`Beta48`	0 | 1 (0)	Beta 48 compatibility.

Error codes

When an esxDOS command fails, BASIC prints one of these messages, which esxDOS reads from the `/SYS/ERRMSG.KO` *file; dot commands report their errors the same way. The code is the number a program gets.*

Code	Message	Meaning
1	O.K. ESXDOS	No error.
2	Nonsense in ESXDOS	Invalid command syntax.
3	Statement END error	Unexpected end of statement.
4	Wrong file TYPE	Wrong file type.
5	No such FILE or DIR	The file or directory does not exist.
6	I/O ERROR	Read or write error.
7	Invalid FILENAME	Invalid file name.
8	Access DENIED	Access denied.
9	Drive FULL	No space left on the drive.
10	Invalid I/O REQUEST	Request beyond the limits of the device.
11	No such DRIVE	The drive does not exist.
12	Too many OPEN FILES	Too many open files.
13	Bad file DESCRIPTOR	Invalid handle.
14	No such DEVICE	The device does not exist.
15	File pointer OVERFLOW	File position overflow.
16	Is a DIRECTORY	The target is a directory.
17	Not a DIRECTORY	The target is not a directory.
18	File already EXISTS	The file already exists.
19	Invalid PATH	Invalid path.
20	No SYS	The function is not available (for example the date and time without `RTC.SYS`).
21	Path too LONG	The path or name is too long.
22	No such COMMAND	Dot command not found in /SYS/BIN or /BIN.
23	File in USE	The file is in use.
24	File is READ ONLY	Read only.
25	Verify FAILED	Verification failed.
26	Loading .KO FAILED	A .KO module cannot be loaded.
27	Directory NOT EMPTY	The directory is not empty.
28	MAPRAM is ACTIVE	MAPRAM mode is active on DivIDE.
29	Drive is BUSY	The drive is busy.
30	Unknown FILESYSTEM	Unknown file system.
31	Device is BUSY	The device is busy.

Sources and links

esxdos.org `www.esxdos.org/`	Official website; download of the current and older versions.
board.esxdos.org `board.esxdos.org/`	Users forum.
esxdos.zxfiles.net `esxdos.zxfiles.net/`	esxDOS software, tips, emulators and links; a ready-to-copy SD card collection with dot commands, games and tools.