ADOS User Manual

Version 1.3 · 25 November 2026 · Adrian Brown

1. Introduction

ADOS is one DOS for the SAM Coupé. It replaces SAMDOS, BDOS (Trinity) and AL-BDOS (Atom Lite), reads and writes MasterDOS disks, and adds PC FAT16 and FAT32 cards and drives. You load it once and every drive you own is available under its own letter.

ADOS fits in one 16K DOS page. Its start-up code is thrown away after boot and the memory is reused for buffers, so ADOS uses no more memory than SAMDOS.

What ADOS gives you

- **One DOS for all your storage:** floppy drives, the Atom Classic and Atom Lite IDE interfaces, and the Trinity SD card interface.
- **Drive letters:** each disk, BDOS record container or FAT partition is a drive with a letter, such as `A:` or `D:`.
- **SAMDOS and MasterDOS compatibility:** disks made by SAMDOS, MasterDOS, BDOS and AL-BDOS work as they are. The error numbers match SAMDOS, so programs that trap DOS errors still work.
- **MasterDOS directories:** subdirectories, paths in file names and date stamps on SAM disks.
- **BDOS records:** a BDOS card or hard drive holds many 800K records, each of which behaves like a floppy disk.
- **PC cards and drives:** a card or drive formatted on a PC as FAT16 or FAT32 is one large drive, with long file names and folders. ADOS reads and writes it, and a PC can read the files ADOS saves.
- **COPY between any drives:** single files, wildcards, and whole disks, with a percentage shown as it goes.
- **SAM disk images on FAT:** an 800K `.MGT` file can be used as a folder or mounted on a drive letter, and works like a real floppy.
- **Real-time clocks:** files you save are stamped with the date and time when a clock is fitted, on SAM disks and FAT drives.

Supported hardware

Device	Notes	Letter
Floppy drive 1 and 2	Internal drives, 800K SAM format	A: and B:
Atom Lite (left and right bays, master and slave)	CompactFlash cards	C: onwards, in the order they are found
Atom Classic (master and slave)	IDE hard drive or CompactFlash	
Trinity	SD card	
Dallas, SAMBUS or Atom Lite clock	Date and time stamps	—

What's new

Version	Changes
1.3	A file loaded in several parts after one header carries on where it left off, instead of starting again each time. This fixes SCREEN$ files loading with the wrong palette.
1.2	Atom Lite cards formatted by AL-BDOS 2.x are found as BDOS cards. AL-BDOS 2.x puts a PC FAT16 boot sector on sector 0 of the card, so ADOS 1.1 took the card for a FAT drive.
1.1	First release. For machine code, as well as everything above: streams (hooks 158, 159, 161 and 192–194) and a direct entry (`CALL &8209`) for programs that have taken the machine over. BDOS programs work as under BDOS: the DVARs are laid out as in BDOS (their address is at offset 0 of the DOS page), `D2:` is the BDOS card, and BDOS's record and sector hooks (134, 149, 156, 157, 160, 162 and 163) are there. With more than one BDOS card, drive 2 is the one used last. Hook 139 (`DVAR`) gives the DVARs: the version is in DVAR 7, DVAR 0 sets the border flash on disk access, and DVAR 48 gives hook errors back to the program. Hook 195 gives a long name (up to 255 characters, with a drive and path) for the next hook that reads a header, so machine code can load, save, open, list and erase files with long names too, through `RST 8` or the direct entry (Appendix A).

About this manual

This manual describes ADOS version 1.3 as it stands on 25 November 2026. Features still to come are collected in Appendix C. Section 5 is the reference for every command; the other sections explain how to use them.

You should already know SAM BASIC. Where ADOS works the same as SAMDOS (for example `SAVE` and `LOAD`), this manual describes only what is different.

2. Getting started

To start ADOS, put the ADOS system disk in drive 1 and press F9, or type `BOOT` and press RETURN. ADOS checks your hardware, shows what it found, and selects a drive. You can then use BASIC as normal.

ADOS can also start from an Atom Lite card with no floppy disk. See MAKEBOOT in Section 10.

The start-up screen

While it starts, ADOS looks for every clock and storage interface it knows about. The screen then shows something like this:

```
     -= A-DOS Version 1.3 (c) 2026 by Adrian Brown =-

Internal Memory: 512K
External Memory: None Present
Clock: AL Clock Present  Date: 20 Nov 2026 14:37:19
Detected: Atom Lite (Right) - Master Size: 159MB Scheme: MBR
  C: FAT16 20MB 'GAMES      '
  D: FAT32 68MB 'WORK       '
Detected: Atom Lite (Right) - Slave Size: 1950MB Scheme: VBR
  E: FAT32 1950MB 'NO NAME    '
Detected: Floppy Drive 1 Size: 800K Scheme: SAMDOS
  A: SAM Disk 800K

Current Drive: A:
```

The sizes, labels and devices depend on your hardware.

- **Clock** shows the clock found, if any, and the date and time it gives (`Not Set` if the clock has no sensible date). See Section 8.
- **Detected** lines list each interface that answered, its size and the scheme: how the medium is laid out (see Section 3).
- The **indented lines** under a device list the drives found on it, with the letter each one was given and, for FAT, the volume label.
- **Current Drive** is the drive used when a command does not name one. ADOS chooses the default drive (normally `A:`). If that drive is missing, it chooses the first letter that has a drive.

Your first commands

1. Type `DIR` to list the files on the current drive.
2. Type `DIR "C:"` to list the files on drive C: without changing the current drive.
3. Type `DIR = "C:"` to make C: the current drive.
4. Type `LOAD "GAME"` to load a file from the current drive, or `LOAD "A:GAME"` to load it from drive A:.
5. Type `SAVE "TEST"` to save the BASIC program in memory on the current drive.
6. Type `COPY "A:*" TO "C:\BACKUP\"` to copy every file on the floppy to a folder on the card.

Taking care of disks and cards

- Do not remove a floppy disk, card or drive while it is being read or written. By default the border flashes while ADOS reads or writes (see Appendix B to turn this off).
- ADOS finds cards and drives when it starts. If you fit a card or drive after booting, reset and boot ADOS again so that it is found.
- A floppy disk, card, BDOS record or disk image can be write protected. ADOS then reports `Protected` when you try to change it.

3. Drives, records and partitions

Every drive in ADOS has a letter, and each letter is one SAM disk, one BDOS record container or one FAT partition. A single card can therefore give you several letters, and you never need to know which interface a file is on, only its letter.

Schemes: how a medium is laid out

When ADOS starts it reads the start of each disk, card or drive to find out how it is laid out. The start-up screen calls this the scheme.

Scheme	What it is	Drives it gives
SAMDOS	An 800K SAM floppy disk made by SAMDOS, MasterDOS or ADOS	One SAM drive
BDOS	A card or hard drive prepared by BDOS or AL-BDOS, holding many 800K records	One drive; use `RECORD` to choose the record
MBR	A PC card or drive with a partition table	One drive per FAT partition
VBR	A PC card formatted as a single volume with no partition table	One FAT drive
GPT	A modern PC partition table	None (not supported)
Unknown	Anything else	None

A card or drive holds either BDOS records or PC partitions, never both.

How letters are given out

- **A: and B:** always belong to floppy drives 1 and 2. `D1:` also means A:. `D2:` means the BDOS card in use, as in BDOS, or B: if there is no BDOS card. A floppy drive that is fitted keeps its letter even with no disk in it, and B: stays free for a disk image when there is no second drive.
- **C: onwards** go to every other drive and partition, in the order they are found: Atom Lite right bay (master, then slave), Atom Lite left bay, Atom Classic master, Atom Classic slave, Trinity. Each FAT partition, and each set of BDOS records, takes the next letter.
- **Up to 8 letters** can be in use at once, counting mounted disk images.

For example, an Atom Classic with two partitions and an SD card as slave gives C: and D: on the hard disk and E: on the card. The letters can change from one boot to the next if you add or remove cards; check the start-up screen, or use `DIR "X:"` to see what a letter holds.

The letters are always given out this way. Only the drive chosen at boot is stored in the DOS itself (DVAR 47, Appendix B), so it can be changed with `POKE` and saved with it.

SAM drives

A SAM drive is an 800K disk: a real floppy, one record on a BDOS card, or a disk image on a FAT drive. SAM drives support everything in this manual, including subdirectories and date stamps.

The directory of a SAM disk normally holds 80 entries. Disks made by MasterDOS with larger directories (up to 780 entries) are read and written correctly.

BDOS records

BDOS and AL-BDOS divide a card or hard drive into records. Each record is an 800K SAM disk and can have a 16-character name. A 2G card holds about 1,200 records.

A BDOS drive always has one record selected, and every command on that drive uses it. Record 1 is selected when ADOS starts.

- `RECORD` lists the records that have names.
- `RECORD 25` selects record 25.
- `RECORD "GAMES"` selects the record named GAMES.

Selecting a record also returns you to its root directory. ADOS reads cards made by both BDOS and AL-BDOS, and the records stay usable by those DOSes.

BDOS programs keep working. ADOS has BDOS's DOS variables and hook codes, and `D2:` is the BDOS card. `PEEK DVAR 25` gives its current record, and `POKE DVAR 25,n` selects record n, as in BDOS. BDOS had one card; with a card in the Atom and another in the Trinity, ADOS's drive 2 and DVARs follow the BDOS card you used last, starting with the first one found. Each card keeps its own record.

FAT drives

A FAT drive is a partition on a card or drive formatted on a PC as FAT16 or FAT32. It is one large drive with folders and long file names, and it is a simple way to move files between a PC and the SAM. ADOS reads and writes FAT drives: `SAVE`, `ERASE`, `RENAME`, `OPEN DIR`, `ERASE DIR` and `COPY` all work on them.

A FAT drive can also hold SAM disk images (Section 6). FAT12, used on very small cards, is shown at start-up but its files cannot be read.

4. File names, directories and paths

A full file name in ADOS is a drive, then a path, then a name: `"D:\GAMES\ARCADE\INVADERS"`. Every part is optional; whatever you leave out comes from the current drive and the current directory.

Drive letters in names

Put the drive letter and a colon in front of a name to use a drive other than the current one:

```
LOAD "B:MYPROG"
SAVE "C:SCORES" CODE 32768,1000
DIR "D:"
```

Upper and lower case letters are treated the same. If the letter has no drive, ADOS reports `No drive`.

Names

- **SAM drives** use names of up to 10 characters, as SAMDOS and MasterDOS do. Case is ignored when ADOS looks for a file.
- **FAT drives** take long names of up to 64 characters, such as `"Manic Miner.bin"` (a longer name is cut to its first 64; a file made on a PC with a longer name can still be reached with a wildcard or its 8.3 name). Case is ignored, and a file can be found by its long name or its short 8.3 name (`MANICM~1.BIN`). A name cannot start with `.` or contain `" * / : < > ? \ |`.
- **The whole name**, drive and path included, can be up to 255 characters, in ADOS's commands and in `LOAD`, `SAVE`, `VERIFY` and `MERGE`: `SAVE "D:\PICS\HOLIDAY SNAPS.SCR" SCREEN$`. The SAM ROM on its own refuses a `LOAD` or `SAVE` name over 14 characters. ADOS takes the longer ones on ROM 3.0; with an older ROM the limit stays at 14. Tape names stay at 10.

Wildcards

Wildcard	Matches	Example	Finds
`?`	Any one character	`DIR "LEVEL?"`	LEVEL1, LEVEL2, LEVELA
`*`	Any characters to the end of the name	`DIR "GAME*"`	GAME, GAMES, GAMEOVER

`DIR`, `ERASE` and `COPY` accept wildcards. `SAVE`, `RENAME`, `OPEN DIR` and `FORMAT` need a plain name and report `Bad file name` if it has wildcards.

Directories

SAM drives hold subdirectories in the same way as MasterDOS, and FAT drives have folders. Each drive remembers its own current directory, so changing directory on D: does not affect A:.

In a path	Means
`\` or `/` at the start	Start from the root directory
`\` or `/` between names	Go down into the directory named before it
`..` or `^`	Go up to the parent directory
`GAMES.MGT`	A disk image on a FAT drive is a folder you can go into (Section 6)

```
DIR = "GAMES"            : REM into GAMES, below the current directory
DIR = "\"                : REM back to the root
DIR = ".."               : REM up one level
DIR = "D:\WORK\SAM"      : REM D: becomes current, in \WORK\SAM
LOAD "\GAMES\INVADERS"   : REM a file in GAMES, from anywhere
LOAD "..\TOOLS\ASM"      : REM up one level, then into TOOLS
```

If a directory in a path does not exist, ADOS reports `Dir not found`. Directories on SAM drives use the MasterDOS layout, so MasterDOS can read them. SAMDOS and BDOS ignore the directory information and show every file on the disk.

SAM files on FAT drives

A file saved to a FAT drive holds only its data, so a PC can read it as it is. The SAM details (type, name, load and run addresses) go in a hidden file beside it with `.info` added to the name: `GAME.BIN` gets `GAME.BIN.info`. ADOS keeps the two together: `ERASE` removes both, `RENAME` renames both, and `DIR` does not list `.info` files.

A file with no `.info`, such as one copied from a PC, loads as CODE. If you do not give an address it loads at 32768. The length always comes from the file's size on the card.

```
LOAD "music.bin" CODE            : REM loads at 32768
LOAD "music.bin" CODE 40000      : REM loads at 40000
```

5. Command reference

ADOS adds the commands below to SAM BASIC, and makes `LOAD`, `SAVE`, `VERIFY` and `MERGE` work on every drive. All of them can be used in a program as well as typed directly. In the syntax lines, *name* is any string expression, and a part in [square brackets] can be left out.

Command	What it does	SAM drive	FAT drive
`LOAD` / `MERGE`	Load a file	Yes	Yes
`VERIFY`	Compare a file with memory	Yes	Yes
`SAVE`	Save a file	Yes	Yes
`DIR`	List files	Yes	Yes
`DIR n`	List drive 1 (A:) or 2 (D2:)	Yes	Yes
`DIR ?`	List files in every directory	Yes	Current folder only
`DIR =`	Change drive and directory	Yes	Yes
`PATH$`	The current drive and path	Yes	Yes
`ERASE`	Erase files	Yes	Yes
`RENAME`	Rename a file or directory	Yes	Yes
`OPEN DIR` / `ERASE DIR`	Make or erase a directory	Yes	Yes
`COPY`	Copy files or whole disks	Yes	Yes
`FORMAT`	Make a blank disk image	—	Yes
`OPEN "X:" TO` / `CLOSE "X:"`	Mount or unmount a disk image	—	Yes
`RECORD`	List or select BDOS records	BDOS drives	Disk images
`BOOT`, F9	Load and run the AUTO file	Yes	Yes

LOAD, MERGE, VERIFY

```
LOAD name [CODE [address]] | [SCREEN$] | [DATA ...]
MERGE name
VERIFY name ...
```

These work as in SAMDOS, and accept a drive and path in the name. A BASIC program saved with `LINE` runs after loading. On a FAT drive, a file with no `.info` loads as CODE (Section 4).

```
LOAD "A:INTRO"
LOAD "\GAMES\INVADERS"
LOAD "D:music.bin" CODE 40000
VERIFY "SCREEN" SCREEN$
```

SAVE

```
SAVE name [LINE n] | [CODE start,length[,run]] | [SCREEN$] | [DATA ...]
```

`SAVE` works as in SAMDOS. The file goes in the current directory of the drive, unless the name includes a path. A file with the same name is replaced. A protected (SAM) or read-only (FAT) file is not replaced, and

ADOS reports `File name used`; so does saving over a folder's name, or over a disk image that is in use.

With a clock fitted, the file is stamped with the date and time. On a SAM drive the stamp goes where MasterDOS keeps it; on a FAT drive it is the file's date, the one a PC shows. Without a clock, FAT files are dated 1 January 1980.

```
SAVE "MYPROG" LINE 10
SAVE "B:SCREEN" SCREEN$
SAVE "D:\WORK\DATA.BIN" CODE 32768,4096
```

DIR

```
DIR
DIR name
DIR n
DIR ?
```

`DIR` lists the files in the current directory of the current drive. With a name, it lists a different drive, or only the files that match the name or a wildcard. To list what is in a directory or a disk image, end the name with `\`: `DIR "D:\GAMES"` lists just the GAMES entry, `DIR "D:\GAMES\"` lists the files in it. `DIR ?` lists every file on a SAM drive, whatever directory it is in; on a FAT drive it lists the current folder.

`DIR n` lists a drive by number, as in SAMDOS and MasterDOS: `DIR 1` is A:, and `DIR 2` is the BDOS card in use, or B: if there is none. `DIR 0` is the current drive; any other number gives `No drive`.

```
DIR "A:"
DIR 1
DIR "D:\GAMES\"
DIR "D:\GAMES\CHAOS.MGT\"
DIR "A:TEST*"
```

On a SAM drive each line shows the entry number, the name, the size in sectors and the type, then the number of files and the space left. On a BDOS drive the heading also shows the record. As in MasterDOS, a CODE file is shown as `C` with its start address and length, and a BASIC program that runs itself shows its auto-run line.

```
A:\
  1 INTRO       12 BASIC 10
  2 GAMES        1 <DIR>
  3 TITLE       49 SCREEN$
  4 MCODE       14 C 32768,6912
 4 files, 731K free
```

On a FAT drive the heading shows the full path, such as `D:\GAMES`. Each line shows the long name and its size: in bytes, in K for files of 64K and over, or `<DIR>` for a folder. Hidden files and `.info` files are not listed.

DIR =

```
DIR = name
```

Changes the current drive, the current directory, or both. See Section 4 for the path rules.

```
DIR = "D:"
DIR = "GAMES"
DIR = "C:\WORK\SAM"
DIR = ".."
DIR = "D:\GAMES\CHAOS.MGT"      : REM into a disk image
```

PATH$

`PRINT PATH$` gives the current drive and path, for example `A:\UTILS` or `D:\GAMES\SAVES`. Inside a disk image it gives only the path inside the image.

ERASE

```
ERASE name
```

Erases a file, or every file that matches a wildcard. Protected (SAM) and read-only (FAT) files are skipped, and so is a disk image that is in use. If nothing matches, ADOS reports `File not found`.

```
ERASE "OLDPROG"
ERASE "B:TEMP*"
ERASE "D:\TEMP\*"
```

RENAME

```
RENAME name TO newname
RENAME DIR name TO newname
```

Gives a file or directory a new name. It stays on the same drive and in the same directory, so give only the name after `TO`. If the new name is already used, ADOS reports `File name used`. `RENAME DIR` is the same as `RENAME`, for MasterDOS users. On a FAT drive the renamed file is dated with the time of the rename.

```
RENAME "PROG" TO "PROG2"
RENAME "D:\WORK\DRAFT.TXT" TO "FINAL.TXT"
RENAME DIR "GAMES" TO "ARCADE"
```

OPEN DIR

```
OPEN DIR name
```

Makes a new directory, in the current directory unless the name has a path. On a SAM disk this is a MasterDOS directory; on a FAT drive it is an ordinary folder a PC can use.

```
OPEN DIR "GAMES"
OPEN DIR "D:\WORK\SAVES"
```

ERASE DIR

```
ERASE DIR name
```

Erases a directory. It must be empty, or ADOS reports `Dir not empty`. On a FAT drive a hidden `.info` file still counts as something in the folder.

COPY

`COPY name TO name`

Copies files between any two drives, SAM or FAT, in either direction. If the target's name is blank or has wildcards, each file keeps its own name. Directories are not copied.

```
COPY "A:CHAOS" TO "D:\GAMES\CHAOS"      : REM one file, new name
COPY "A:*" TO "D:\BACKUP\"              : REM every file, same names
COPY "D:\PICS\*.SCR" TO "A:"            : REM matching files
COPY "D:README.TXT" TO "D:\DOCS\"       : REM same drive, another folder
```

While it copies, COPY shows each file's name and how far it has got, from 0% to 100%. The figure counts every sector read and written, so on a small file it can reach 99% early and then stay there until the file is finished.

As in MasterDOS, a file bigger than free memory is copied in several passes, so any file can be copied whatever its size. COPY needs at least 32K of free memory in one block; with less it reports `No memory`. The more memory is free, the fewer passes it takes.

The copy keeps the file's type and addresses. The copy is dated when it is made, as if it had just been saved. A PC file copied from one FAT drive to another gains a `.info`.

Whole disks. If one side is only a drive letter and that drive is a SAM disk, COPY copies the whole disk, every track and sector, and shows a percentage as it goes. The target is overwritten completely.

```
COPY "A:" TO "B:"                 : REM floppy to floppy
COPY "A:" TO "D:\BACKUP.MGT"      : REM floppy to a disk image (made if it isn't there)
COPY "C:" TO "D:\RECORD5.MGT"     : REM the current BDOS record to a disk image
COPY "D:\BACKUP.MGT" TO "A:"      : REM disk image to a floppy (it must be formatted)
```

FORMAT

`FORMAT name`

Makes a new, empty SAM disk image (819,200 bytes) on a FAT drive. If the name is already used, ADOS reports `File name used`. FORMAT does not format real floppies; on a drive that isn't FAT it reports `Bad device`.

```
FORMAT "D:\WORK.MGT"
```

OPEN "X:" TO and CLOSE "X:"

`OPEN "X:" TO name`
`CLOSE "X:"`

`OPEN` gives a disk image its own drive letter until `CLOSE` or reset. The letter must be free, or ADOS reports `Bad device`. If the closed drive was the current drive, ADOS goes back to the default drive. `OPEN #` and `CLOSE #` (streams) are still the ROM's.

```
OPEN "B:" TO "D:\GAMES\CHAOS.MGT"
LOAD "B:chaos"
CLOSE "B:"
```

RECORD

```
RECORD
RECORD n
RECORD name
```

Works on the current drive, which must be a BDOS drive; otherwise ADOS reports `Bad device`. `RECORD` on its own lists every record that has a name, with its number. A name is up to 16 characters, and ends early at a 0 byte (AL-BDOS ends names with zeros, BDOS pads them with spaces). `RECORD n` or `RECORD name` selects a record; case is ignored. If the record does not exist, ADOS reports `No record`. Select the drive first with `DIR =`:

```
DIR = "C:"
RECORD
RECORD 12
RECORD "UTILITIES"
```

On a FAT drive, `RECORD "CHAOS.MGT"` does the same as `DIR = "CHAOS.MGT"`, and `RECORD 0` leaves the image (but not a mounted one: use `CLOSE`).

BOOT and F9

F9 loads and runs the file whose name starts with `AUTO` on the current drive, in the current directory. If there isn't one, nothing happens. `BOOT` does the same once ADOS is loaded, but reports `No AUTO file` if there isn't one. The AUTO file must be BASIC, CODE or SCREEN$; BASIC programs run from their auto-start line.

6. Disk images

A FAT drive can hold SAM floppy disk images. Software that needs a SAMDOS or MasterDOS disk runs from an image unchanged. There are two ways to use one:

- **As a folder.** Go into the image with a path: `DIR = "D:\GAMES\CHAOS.MGT"`. While inside it, the drive works as a SAM floppy. `..` from the image's root goes back to the FAT folder that holds it.
- **Mounted on its own letter.** `OPEN "B:" TO "D:\GAMES\CHAOS.MGT"` makes B: a SAM floppy until `CLOSE "B:"`. This suits software that only uses certain drives, and games on more than one disk. A mounted drive stays inside its image.

An image can also appear in the middle of a path: `LOAD "D:\GAMES\CHAOS.MGT\chaos"`.

Which files are images

- An image is any file of exactly 819,200 bytes, whatever its extension (`.MGT`, `.DSK`, or none): the raw 800K disk in MGT order. EDSK and SAD files are not read.
- An image need not be in one piece on the card, but must be in no more than 12 pieces, or ADOS reports `Fragmented`. Copy it to another drive and back, or defragment the card on a PC.
- An image that is read-only on the FAT drive reports `Protected` when written to.
- An image that is mounted, or that a drive is inside, cannot be erased or saved over. `RENAME` is still allowed.

Use `FORMAT` to make a blank image, and `COPY` to copy a whole disk to or from one (Section 5).

7. Interfaces

ADOS finds every supported interface by itself when it starts; there is nothing to set up. This section gives the details that matter when you fit or swap hardware.

Interface	Media	Up to	Notes
Floppy drives	800K SAM disks	2 drives	Drive 2 cannot be used if an Atom or the Atom Lite right bay is fitted
Atom Classic	IDE hard drive or CF	Master and slave	Drives are not hot swappable
Atom Lite	CompactFlash	2 bays × master and slave	Left bay replaces floppy 1, right bay replaces floppy 2
Trinity	SD or SDHC card	1 card	Reboot after swapping the card

Floppy drives

Floppy drives read and write 800K SAM disks (80 tracks, 2 sides, 10 sectors of 512 bytes), the same format as SAMDOS and MasterDOS. The drive motor stops after a short time with no activity.

Atom Classic

The Atom Classic is a 16-bit IDE interface for a hard drive or CompactFlash adapter, with a master and a slave. It uses the ports of floppy drive 2. Drives prepared by BDOS hold BDOS records, read in the byte

order BDOS used so that BDOS can still read them. Drives formatted on a PC hold FAT partitions. ADOS uses LBA addressing, so drives larger than 8G can be used for FAT32.

Atom Lite

The Atom Lite is an 8-bit CompactFlash interface with two bays. The right bay uses the ports of floppy drive 2 and the left bay those of floppy drive 1. Each bay can hold a master and a slave card. Cards prepared by AL-BDOS hold BDOS records, and AL-BDOS can still read them after ADOS has used them. Cards formatted on a PC hold FAT partitions.

Trinity

Trinity is an SD card interface. ADOS reads and writes cards prepared by BDOS, and PC-formatted FAT cards. BDOS supported only SD cards up to 2G; ADOS also supports SDHC cards (4G to 32G), which is what most FAT32 cards are.

> SDHC support has not yet been tested on real hardware. If an SDHC card is not found, try a 2G SD card.

Using cards made on a PC

1. Partition the card or drive with an MBR ("Master Boot Record" in the PC's disk tool), not GPT. A card formatted as one volume with no partition table (a "superfloppy") also works.
2. Format each partition as FAT16 or FAT32. Do not use exFAT or NTFS.
3. Copy your files to it. Files saved by ADOS keep their SAM type through their `.info` files; other files load as CODE.

8. Real-time clocks

When a clock is fitted, ADOS stamps every file you save with the date and time, on SAM and FAT drives alike. There is nothing to set up: the clock is found when ADOS starts, and the start-up screen shows which one it is and what it says the date and time are, for example `AL Clock Present  Date: 20 Nov 2026 14:37:19`.

Clock	Start-up screen shows
Atom Lite clock (AL+)	AL Clock Present
SAMBUS clock	SAMBUS Clock Present
Dallas clock	DALLAS Clock Present
No clock	No Clock Present

If more than one clock is fitted, ADOS uses the first it finds, in the order of the table. On a SAM drive the date stamp is stored where MasterDOS keeps it, so MasterDOS shows the same date for the file. On a FAT drive it is the date the file was written, which a PC shows; the dates it was created and last opened are left empty. If the clock is not set, or gives a date that cannot be right, the start-up screen shows `Not Set`, SAM files are saved without a date and FAT files are dated 1 January 1980.

9. Error messages

ADOS reports errors with the same numbers as SAMDOS, so a program that traps errors with `ON ERROR` sees the code it expects. Errors 114 onwards are new in ADOS.

No.	Message	Usual cause
81	Nonsense	A command's syntax is wrong, for example a missing `TO`
82	No memory	COPY has less than 32K of free memory in one block
83	Bad statement	Something extra after the command
85	Disk error	The drive reported a fault
87	No disk	No disk or card in the drive
89	Bad file name	A bad name, wildcards where they aren't allowed, or a name too long
91	Bad device	The wrong kind of drive for the command, or a letter already in use
93	Verify failed	VERIFY found a difference
94	Wrong type	The file isn't the type asked for
101	No AUTO file	BOOT found no `AUTO*` file
103	No drive	No drive has that letter
104	Protected	The disk, card or image can't be written to
105	Disk full	The drive is full
106	Dir full	No free directory entries
107	File not found	No file matches the name
108	End of file	A file ended sooner than expected
109	File name used	The name exists, or the file is protected
113	Bad hook	A program called a hook ADOS doesn't know
114	No record	The BDOS record doesn't exist
115	Not a SAM disk	The disk isn't in a format ADOS can read, such as FAT12
116	Dir not found	No directory with that name
117	Dir not empty	ERASE DIR on a directory that still holds files
118	Fragmented	A disk image is in more than 12 pieces on the card
119	File not open	A stream hook with no file open (Appendix A)

10. Utilities

The ADOS disk comes with utility programs. Each is a program of its own, not part of ADOS. It runs from BASIC under any DOS.

Utility	What it does
BDOS2FAT	Turns a BDOS drive into a FAT32 drive, with every record in use as an 800K disk image file
MAKEBOOT	Makes an Atom Lite card boot ADOS by itself at power on, with no floppy disk

BDOS2FAT

BDOS2FAT turns a BDOS drive into a FAT32 drive, on the SAM itself. Every record in use becomes a disk image file in the root folder, which ADOS, SimCoupe and a PC can all use. The drive can be an Atom Classic or Atom Lite hard drive or card, or a Trinity SD card.

- **What you get.** A FAT32 drive with an MBR and one partition covering the whole drive, and one 800K image (MGT order, the same as an `.MGT` file) for each record in use. Open one with `DIR = "C:\0012 GAMES.MGT"` or mount it with `OPEN "B:" TO ...` (Section 6).
- **Which records.** A record is converted if its directory holds at least one file, even if the first files were erased. A name on its own doesn't count: BDOS names every record it formats, so a card can have thousands of named records that are empty (`blank302`, `blank303`, ...). Records with no files are left out, and their space becomes free space.
- **Names.** You choose either the record's number and name (`0012 GAMES.MGT`, `0007.MGT` for a record with no name), or the name only (`GAMES.MGT`, `Record 7.MGT` for no name). With the name only, a name already used by an earlier record gets the record's number added: `GAMES (37).MGT`. You also choose `.MGT` or `.DSK`. The characters `" * / : < > ? \ |` become `_`, spaces at either end are removed, and the SAM's £ and © stay as £ and ©.

Before you start

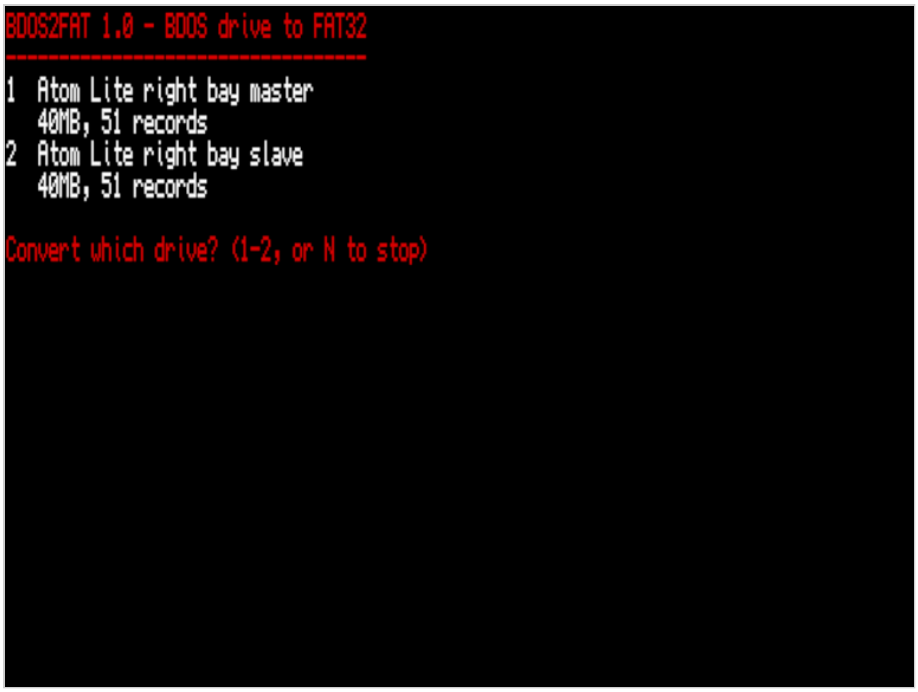
- **Back up the drive.** BDOS2FAT rewrites the whole drive, and BDOS can't use it afterwards. On a PC, take an image of the card or drive first.
- **Free records.** The FAT goes at the start of the drive, so the records there are moved into records that aren't in use: usually one or two. BDOS2FAT says if there aren't enough before it changes anything.
- **Size.** The drive must be at least 33MB, the smallest a FAT32 drive can be.

Running it

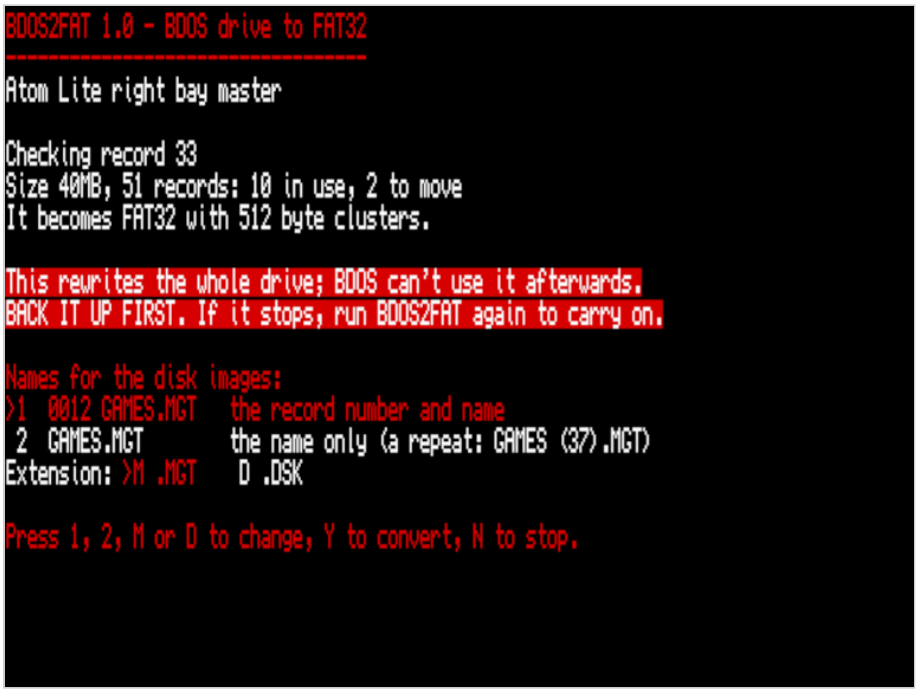
BDOS2FAT is on the ADOS disk. Load it from BASIC, and it starts as soon as it has loaded:

```
CLEAR 32767: LOAD "BDOS2FAT" CODE
```

It changes to screen mode 3 and lists the BDOS drives it finds. Press a drive's number, or `N` to stop. It looks at the drive, only reading, then shows its size, the records in use, how many must move, and the FAT32 cluster size, with the choices for the names. Press `1` or `2` for the names, `M` or `D` for the extension, then `Y` to convert the drive (and `Y` again when asked if you are sure), or `N` to leave it unchanged.



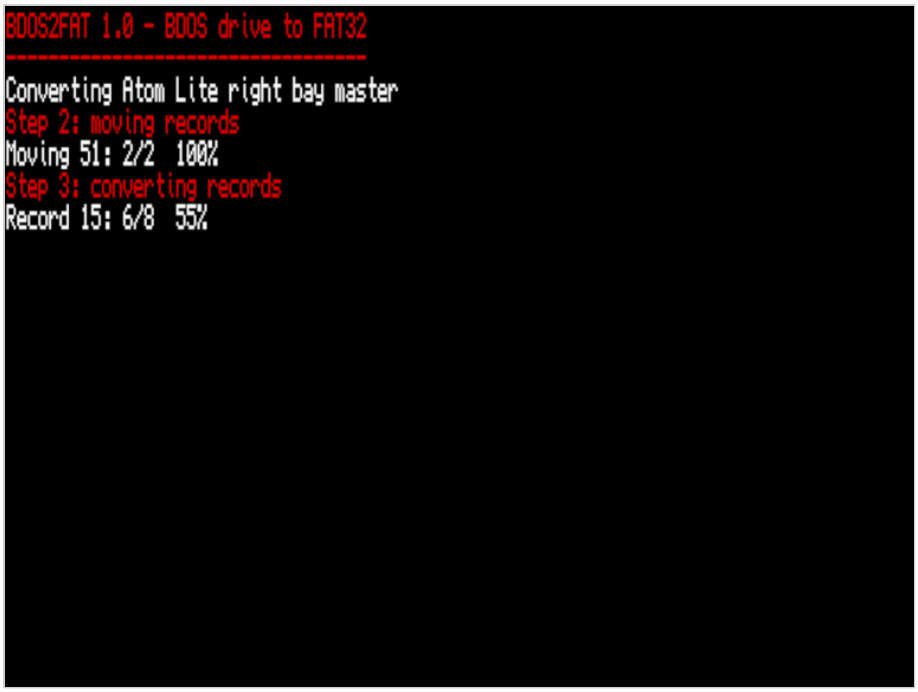
The BDOS drives found (here an Atom Lite master and slave)



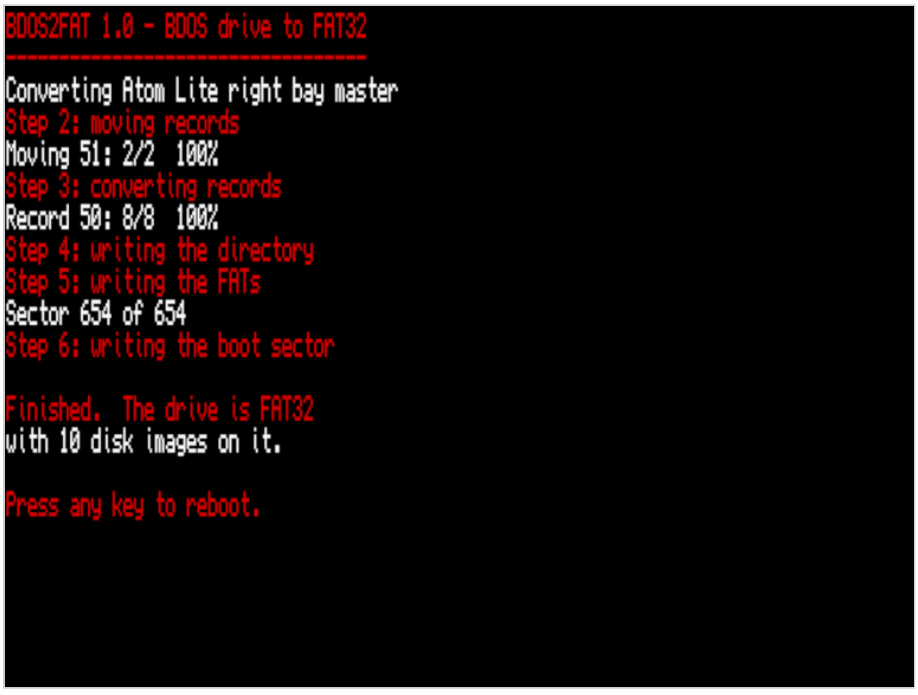
The drive, what will happen, and the choices for the names

Step	What it does
1	Looks at the drive and asks before going on. Nothing is written until you have answered Y twice.
2	Copies the records in the way of the FAT (and a partial last record) into free records
3	Converts each record in use where it is
4	Writes the root folder, using the BDOS record names
5	Writes the two FATs
6	Writes the boot sector, then the MBR (this makes it a FAT32 drive)

Step 3 takes most of the time: allow roughly 15–40 seconds a record. A progress line such as `Record 12: 3/9 45%` means record 12, the 3rd of the 9 records to convert, 45% of the way through it. When it says `Finished`, it asks you to press any key, and then reboots the SAM. It doesn't go back to BASIC, because the DOS that is loaded still thinks the drive is a BDOS drive. It stops the same way after a disk error.



Part way through step 3: record 15, the 6th of 8 to convert, is 55% done



Finished, waiting for a key to reboot

If it stops

A power cut, a reset or a disk error part way through doesn't lose anything. BDOS2FAT keeps a note in the drive's first sector of how far it has got. Run it again: it lists the drive as `Conversion not finished`, and carries on from where it stopped. Until it has finished, neither BDOS nor ADOS can use the drive properly, so don't write to it with anything else.

What it does to the drive

A BDOS record already holds exactly the sectors of a disk image, in a different order: BDOS keeps side 1's tracks, then side 2's, while an image interleaves the sides. BDOS2FAT lays the FAT32 drive out so that each record's sectors become its file, so the records stay where they are and only their tracks are put in order. On an Atom Classic it also puts the bytes of each sector in PC order. The clusters are as big as FAT32 allows on the drive, up to 32K, which keeps the FAT small: fewer records need moving, and ADOS finds free space faster. Each image is exactly 25 clusters of 32K, so no space is wasted.

MAKEBOOT

MAKEBOOT makes an Atom Lite card boot ADOS by itself when the SAM is switched on or reset, with no floppy disk and no Liaise. It needs the Atom Lite boot ROM (Edwin Blink's AL BOOT ROM, from World of SAM), which boots from a card whose sector 0 holds a boot loader. MAKEBOOT puts ADOS's loader there and ADOS itself in sectors that belong to no file system.

- **Sector 0** gets the loader, in the bytes the layout leaves free. An MBR's partition table, or a FAT boot sector's BPB, stays as it is, so a PC still sees the card as it was.
- **38 sectors of ADOS** go in the first run of free sectors: the gap before the first partition, or the reserved sectors of a FAT volume. On a BDOS card they go in a record that has never been formatted, and the card's own boot sector is saved after them.
- **Nothing else is changed.** No file, FAT or directory is touched, and no formatted BDOS record. If there is no room, MAKEBOOT says so before it writes anything.

Card	Result
Used on a PC (MBR, FAT16 or FAT32)	Works if there is a gap before the first partition (PC tools usually leave 2048 sectors)
Converted by BDOS2FAT	Works with the BDOS2FAT on this release's disk, which keeps 64 or more reserved sectors. A card converted by an earlier one may keep only 32, and is then refused
FAT32 with no partition table	Usually refused: its reserved sectors are too few
AL-BDOS (or a renamed one)	Works if one of its records has never been formatted. Formatting that record in BDOS would remove ADOS. If every record is formatted it is refused
Plain BDOS, with no boot sector	Refused: the ROM can't boot it

MAKEBOOT is on the ADOS disk. Load it from BASIC, and it starts as soon as it has loaded:

```
CLEAR 32767: LOAD "MAKEBOOT" CODE
```

It lists the Atom Lite drives that have a card. Press a drive's number, or N to stop. It reads the card (only reading), shows where ADOS will go, and asks. Press Y to write it, or N to leave the card unchanged. It writes ADOS first and checks it, then writes sector 0 last, so a card is never left with a boot sector that has nothing to load. Run MAKEBOOT again to update ADOS on the card.

Booting from the card

Fit the AL BOOT ROM and switch on or reset. The card's boot sector loads ADOS and starts it, then the SAM goes to BASIC, so DIR, LOAD and the rest work. At this point in the ROM's start up the screen can't

be set up, so ADOS shows no start-up screen as it does from a floppy.

- **Drives.** The default drive is the floppy (`A:`), as after a floppy boot. Type the card's drive letter to use the card, for example `C:`.
- **F9** doesn't restart the SAM. It loads the `AUTO*` file from the current drive, and reports `No Disk` if that is the floppy and it is empty. To boot the card again, reset.
- **Floppy or another card.** Hold SPACE at reset to boot from a floppy instead, or 1 to 4 to pick a card, as the ROM does for any DOS.
- **BDOS cards.** Hold B while resetting to boot BDOS instead of ADOS. With no key held, ADOS loads.
- **BDOS menu programs**, such as the record selector in a card's first record, don't work under ADOS: they use BDOS's private memory, and the list shows rubbish. Use them under BDOS (hold B). If you have run one under ADOS, reset the SAM, as it may have changed ADOS's memory.

Appendix A. DOS hook codes

Machine code programs call ADOS through the ROM's DOS hook codes, as with SAMDOS: put the arguments in the main registers, then `RST 8` followed by the hook code. Results come back in the alternate registers. Errors end in a BASIC error, unless DVAR 48 is set (Appendix B): then carry is set in F' and A' holds the error number. Hooks from 128 to 194 that ADOS does not list below do nothing. Hook 195 gives a long name for the next hook that reads a header (below). A code outside 128–195 gives `Bad hook`.

Code	What it does	Arguments	Results
128	Load and run the `AUTO*` file if there is one (F9)		
129	Find a file and give its header	IX = header with type and name	Header at &4B50
130	Load the file found by 129	HL = address (section C), C = 16K pages, DE = bytes mod 16K	
131	Verify it	As 130	
132	Save	IX = full 48-byte header	
134	Verify sectors (BDOS)	As 162	
135	Select the default drive (starts ADOS at boot)		
136	As 128, but error 101 if there is no `AUTO*` file		
139	DVAR: the address of DOS variable n (BASIC's `DVAR n`). Every DOS has this hook	n on the calculator stack	The address on the calculator stack
142	PATH$		String on the calculator stack
145	Set the current drive and directory (`DIR =`)	A = page, DE = address, BC = length of the name	
149	Write a sector	As 160	
156	Select a record on the BDOS card (BDOS)	A = 0: HL = record number; A = 1: HL = address of a 16-character name	
157	Verify a block from the stream (BDOS)	As 161	As 161
158	Open a file as a stream	IX = header with type and name	Header at &4B50, DE':HL' = length
159	Read one byte from the stream		A' = the byte
160	Read a sector	A = drive (1, 2 or a letter), D = track, E = sector, HL = address 16384–65535	
161	Read a block from the stream	A = 16K pages, DE = bytes mod 16K, HL = address (section C)	C':DE' = bytes read, B':HL' = next free byte

162	Read sectors (BDOS)	A = drive, D = track, E = sector, C = page, HL = offset, IX = how many	
163	Write sectors (BDOS)	As 162	
165	Print a directory listing	IX = header with the name or pattern	
166	Erase a file	IX = header with the name	
192	Move the stream to a position	DE:HL = position	
193	The stream's position		DE':HL' = position
194	Close the stream		
195	Long name for the next hook that reads a header	A = page, DE = address, BC = length of the name (0 cancels)	

Long names. A header has room for a 14-character name. Hook 195 gives a longer one, up to 255 characters, with a drive, a path and a FAT long name allowed as in `LOAD`. The next hook that reads a header (129–132, 158, 165 or 166) uses it in place of the header's name, once. The arguments are those of hook 145: A = the page, DE = the address as if that page were in section C. With BASIC's paging, a name at 16384–32767 is page 0 at the address + &4000, one at 32768–49151 page 1 at the same address, and one at 49152–65535 page 2 at the address − &4000. Call 195 just before that hook: anything else that reads a name in between, or any BASIC error, cancels it. It also works through the direct entry.

BDOS hooks. Drive 1 is A:, drive 2 is the BDOS card in use (B: if there is none), and a letter such as "C" can be given instead. The sector hooks work on any SAM drive: a floppy, the current BDOS record, or a disk image. For 160 and 149 the address is read as BASIC's memory is paged (16384–65535, pages 0–2). A track and sector that isn't on the disk gives `Disk error`. `Docs/Hooks.md` lists the BDOS DVARs and the details.

158, 159 and 161 have the same numbers and arguments as in MasterDOS and BDOS. A stream reads one file a piece at a time, from any position, on any drive, including inside a disk image. Positions count data bytes only, so position 0 is the first byte of the data on every drive. Streams are read only. A stream hook with no file open gives `File not open`, except 194 (close).

Is ADOS loaded? Codes 192 and up are ADOS's own, and MasterDOS crashes on unknown codes above 174, so check first. DOSFLG (&5BC2) is 0 if no DOS is loaded. Otherwise read DVAR 7 through hook 139, which every DOS has (from BASIC, `PEEK DVAR 7`): ADOS holds version × 10 + 50 there, 63 for ADOS 1.3. To be certain it is ADOS and not another DOS using the same number, also check for the letters `ADOS` at offset &0212 of the DOS page, followed by the version (major, minor). `Docs/Hooks.md` has the code for both.

Direct entry (from 1.1). A program that has taken the machine over (ROM 0 paged out, the system variables reused, its own interrupts) can't use `RST 8`. Instead it pages the DOS page into section C and calls offset &0209 in it, with the hook code in the byte after the call: `CALL &8209` / `DB 161`. The arguments and results are the hook's, in the main registers. An error comes back to the program with carry set and A = the error number. Hooks 132, 145, 158, 161, 192–194 and 195 can be called this way; for 161, C gives the page to read into. The paging and interrupt state are as they were. Interrupts stay on except while a piece of data is copied, so an IM2 handler kept in the page after the DOS page keeps running. `Docs/Hooks.md` has the details.

Seeking reads only the sector the new position is in. On FAT it follows the cluster chain without reading the data. A SAM file whose sectors are in one run (as SAVE makes them when there is room) goes straight to the right sector. Other SAM files are read through, because each sector holds the link to the next.

The ADOS source includes `Docs/Hooks.md`, with full details and a worked example.

Appendix B. Memory use and settings

ADOS lives in one 16K page, paged in at &4000–&7FFF while it runs.

Address	Contents
&4000–&4001	The address of the DVARs as seen in section C (&8220), as in BDOS (put there at boot; on disk these bytes are the file header)
&4003–&4005	COPY progress
&4009–&40FF	Boot loader; after boot, the open stream's state and other buffers
&4100	BOOT signature, then shared routines
&4200	Jump block (hooks, syntax, NMI), then the direct entry at &4209
&4210	Error table pointer, then `ADOS` and the version at &4212, then the direct entry's variables
&4220	DVARs: settings saved with the DOS
After the DVARs	Resident code
After the resident code	Start-up code, reused as about 1.5K of workspace after boot
&7DA0–&7F9F	Sector buffer (512 bytes)
&7FA0–&7FFF	Stack

The code that finds the drives is only needed at start-up, so it runs from a spare memory page and takes no room in the DOS page.

In the system page ADOS keeps a 32-byte routine at &4BA0, and points the ROM's RST8V vector (&5AEE) at it. ROM 3.0 refuses a `LOAD` or `SAVE` name over 14 characters with error 18, and its error handler calls RST8V first. For that error the routine passes the whole name to ADOS (hook 195), then lets the ROM carry on as if the name were 14 characters. Every other `RST 8` goes straight through. ADOS puts the vector and routine back whenever the DOS is initialised (hook 135).

DVARs

The DVARs at &4220 hold the settings below. DVAR 0–45 are laid out as in BDOS, so BDOS programs find what they expect, and DVAR 0–7 also mean the same as in SAMDOS and MasterDOS. Changing a DVAR with `POKE DVAR n,value` and saving the DOS file keeps it for the next boot.

DVAR	Address	Setting	Default
0	&4220	Border flash while reading and writing: the colour is (DVAR 0 AND 7) AND the sector number. 0 turns it off	7

1–45	&4221–&424D	BDOS's, describing the BDOS card in use: DVAR 25 is its current record (POKE it to select another), 23 the number of records, 39–44 the date and time. `Docs/Hooks.md` lists them all	—
7	&4227	Version × 10 + 50	61
46	&424E	Drive timeout, in units of about 1.5 seconds	3
47	&424F	Drive selected at boot	"A"
48	&4250	Not 0: a failed hook gives the error back to the machine code program (carry set, A' = the error) instead of a BASIC error. For programs only; see the hook guide	0

While DVAR 48 is set, commands typed in BASIC and the ROM's own LOAD and SAVE also fail without a message, so a program should set it only around its own calls.

Appendix C. Planned features and known limits

Known limits in this version

- **FAT12** drives are shown at start-up but cannot be read. GPT partition tables are not supported.
- **Dates on FAT:** DIR does not show dates, and RENAME gives a FAT file the time of the rename.
- **Single-drive copies:** MasterDOS asks you to swap disks when copying from one floppy to another in the same drive. ADOS does not, so copy through a second drive or a disk image.
- **Inside a disk image,** the DIR heading and PATH$ show only the path inside the image, not the FAT folder and image name.
- **New cards and drives** are found only when ADOS starts.
- **Streams** are read only, and don't notice a floppy being changed while a file is open (close the file, change the disk, and open it again).
- **Long listings:** a long wildcard COPY or DIR stops at the ROM's `Scroll?` prompt when the screen fills, as in SAMDOS.

Planned

Feature	Notes
`DEVICE`	Move a drive to another letter and save the setting, without POKE
`FORMAT` for floppies and BDOS records	Today FORMAT only makes disk images
`PROTECT` / `HIDE`	Protect or hide a file
`READ AT` / `WRITE AT`	Read or write a single sector
`OPEN #` and `POINT` on files	BASIC streams to files, as in MasterDOS, and writing through a stream
Dates in DIR	Show each file’s date in DIR listings
ATAPI CD-ROM	On the Atom Classic, read only (ISO 9660)
RAM discs	If memory allows