

APT Partition Editor and Utilities

Version 1.0

User's Manual

Introduction

The Atari Partition Table (APT) is a design standard (drafted by KMK) intended to provide compatibility among compliant Atari hard disk interface hardware. Any hard disk interface whose firmware, driver or BIOS follows the APT specification will be compatible with the APT partition editor and associated utilities, such as DEVINFO and MOUNT. Currently, the SIDE and MyIDE interfaces have APT drivers which work with the APT software. It is hoped that IDE Plus and the KMK/JZ interface will soon have APT support, allowing for hard disk media to be completely interchangeable between all these compliant interfaces.

Currently, the SIDE and MyIDE APT drivers are reliant on the use of SpartaDOS X, and it follows that the APT partitioning software also currently requires the use of this DOS. When bootable interfaces such as IDE Plus adopt the APT standard, APT software will be available to run under different disk operating systems.

Features and Utilities

The current APT software implementation (written by Flashjazzcat) provides the following features:

- Support for 127 partitions per disk (although there is theoretical support for 65,535)
- Support for 15 drives ("A:" to "O:")
- Support for sector sizes of 128, 256 and 512 bytes
- Partition locking and deactivation
- Dynamic mounting and un-mounting of partitions
- Partition naming

Software may be supplied on an ATR disk image, or on the CAR: device in an SDX ROM distribution. In either case, the following programs are provided:

- FDISK.COM (partition editor, on support disk)
- PARTINFO.COM (quick partition display utility)
- DEVINFO.COM (physical disk info tool, written by KMK)
- MOUNT.COM (dynamic partition management utility)
- HDDINFO.COM (provides basic information about physical disks)

Installation and Disk Partitioning

Before using a disk, it must first be initialized. Attach the hard disk to the interface, connect the interface to the Atari, and power on the machine. SpartaDOS X will boot and the APT BIOS will report “Invalid partition table”. This means the hard disk has not been initialized (or that there was an error reading the partition table). Type `FDISK` at the command prompt, and the partition editor will load and display the following screen:



FDISK is simple to use: it is navigated using the keyboard, largely via the Esc, Return, and the cursor keys. You'll note that the “Disk” menu is already highlighted on the menu bar at the top of the screen. Press <Return> to open the menu, then choose “Open disk” (just press <Return> again, since it is the first item on the disk menu):



The next window to open will display the physical disk(s) that the driver has previously detected. Use the cursor keys to highlight the disk you wish to partition, and press <Return>:



If the disk hasn't already been prepared, FDISK will then ask you to specify the amount of space you wish to allocate for the Atari partitions. You'll also have the chance to create an (optional but recommended) FAT32 partition, which can be used with the SIDE loader.



Respond "Y" to "Partition now: (y/n)?" to prepare the card. **NOTE: all existing information on the card will be lost, so please use this option with care. FDISK obliterates the existing boot sector or MBR partition table.**

You'll then get the chance to review the geometry of the hard disk.

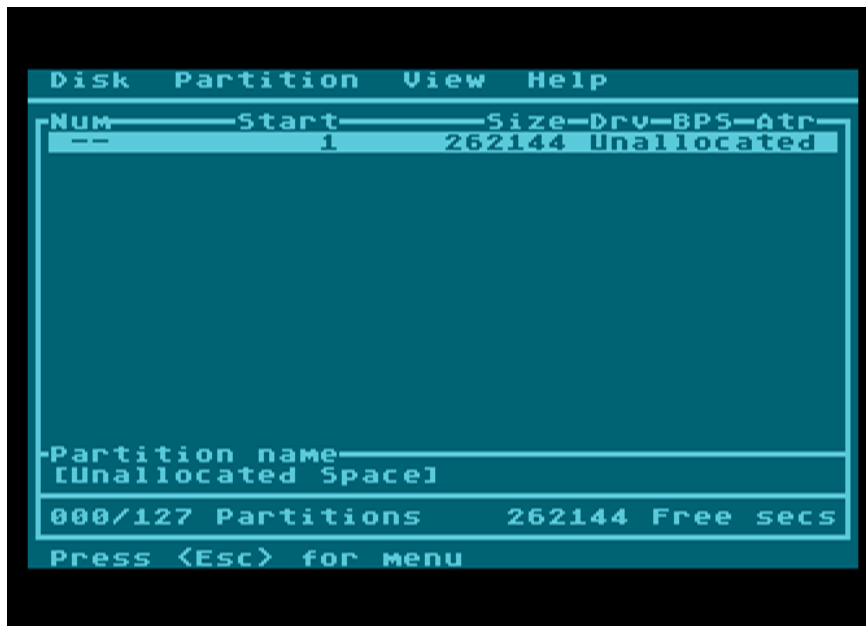


Just press any key to advance from this screen. FDISK will then ask whether you require a FAT32 partition on the disk (always the first on the physical disk). If you select "Y", the FAT32 partition will occupy all the space on the disk *not occupied by APT partitions*. Whether or not you specify that a FAT32 partition should be created, in the next dialogue you will be asked for the size of the APT segment on the disk:



"Free space" represents the total size of the physical hard disk in megabytes. If you chose to create a FAT32 partition in the previous step, that partition will consume all the disk space you don't allocate for Atari partitions. Note that the minimum size of the FAT partition is 32MB. If you chose not to have a FAT32 partition, you may allocate all of the free space to the APT.

Once you have specified the size of the APT, the MBR will be written to the disk and you'll be presented with the Atari partition table editor as shown in the next screenshot.



You will note that the menu bar has expanded to show three items. To activate the menu, press <Esc>. Use the cursor keys to choose the menu you wish to open, then press <Return>. You can then scroll through the unfurled menus with the left and right cursor keys.

To create a partition, press <Esc>, move the highlight to "Partition", press <Return>, highlight "Create New", then press <Return> again:



You can also just press the letter "C" without opening any menus (note that all these keyboard shortcuts are shown on the pull down menus). In any case, a numeric input field will open up in the first (empty) entry in the partition table, waiting for you to specify the size of the partition in sectors. The default is 65535 sectors, and if you want to accept this value, just press <Return>, otherwise you can edit the value using the usual editing controls. After you press <Return>, the new entry will appear in the table will appear, and the free space following it will shrink accordingly.

With the new partition highlighted, you may then specify a drive letter, either by pressing “V” or selecting “Set Drive” from the partition menu. You’ll get a drop-down list box. Just select the drive letter you want and press <Return> (note that if the partition already has a drive letter, this is shown near the top of the list and is already highlighted. You can also remove the drive letter by choosing the “—” selection):



Partitions are 512 bytes per sector (BPS) by default, but you may also change this by pressing “S” or choosing the appropriate menu option. You get three options: 128, 256 and 512bps, corresponding to single, double and quad density respectively:



Partitions can have names up to 39 characters long. Pressing “N” or selecting “Edit Name” on the menu will allow you to type the partition name. Press <Return> when done.

You may also set the attributes (in the “Atr” column) via the menu or using the hotkeys.

“A” means the partition is active (it will be mounted at startup providing it is assigned a drive letter).

“L” means the partition is write-locked, so it will not be possible to write to it (it will be read-only).

“B” means the partition is the boot partition. There can only be a single boot partition, and this is the partition that AUTOEXEC.BAT will be read from when the computer boots. It will also become the default drive. This option is a convenient way of – for example – booting from drive C and leaving drives A and B free for optional SIO devices (similar to a traditional PC setup). Just create a partition with drive letter C and set the boot flag.

Here’s a typical partition table with three quad-density partitions assigned to drives C, D and E. Drive C is the boot drive.



When you have the partition table the way you want it, choose Disk->Write Table or press <Ctrl+W>. FDISK makes absolutely sure you want to commit the changes to disk.

After writing the changes, you may close the partition table with Disk->Close:



Finally, to exit FDISK and return to the SDX command prompt, chose Disk->Exit. The partition table will be automatically refreshed from the hard disk, and the partitions you have created will immediately become available. You must then format them with the SDX disk formatter (type "FORMAT" at the command prompt).

The illustration below shows the SIDE driver booting under SpartaDOS X, with a fully partitioned hard disk:



```
Ultimate clock installed
SIDE Driver v1.0 (c)2011, FJC
GENERIC
Drive(s): C,D,E
Boot drive: C:

    SpartaDOS X 4.45 8-11-2011
    Copyright (C) 2011 by FTE & DLT

D3:■
```

Partition Table Maintenance

It's likely that after initialising a card and using it, you'll want to add or remove partitions at a later date. Just start FDISK again, select the IDE disk, and you'll see the partition table in the editor. You can continue to add partitions or change the size of existing ones. Note, however, that deleting or resizing existing partitions will almost certainly result in data loss if they already contain a filing system (i.e. they are formatted) and there are files and folders on the partition. For this reason, FDISK asks you repeatedly if you are sure you wish to write changes to the disk.

While editing the partition table, you may decide that you wish to discard all the changes you have made since the table was read from the disk. You can accomplish this with Partition->Undo all, although the result is identical to simply closing the disk, choosing NOT to save changes, and then opening the disk again.

If you have named partitions, you may find it helpful to list the partition table by name. You can do this by choosing View->Show names, which simply toggles the partition name view. You can still create, edit and delete partitions in name view.



Table Import/Export is a useful facility you can use to simplify the creation of similar partitioning structures on many disks. With an existing partition table open in the editor, select Disk->Export table and a file selector will be displayed:



You can move between the file list and the input line with the <Tab> key. Use <Ctrl+L> to followed by “A” through “O” to select a drive letter. When you’ve logged a suitable location for the exported partition table, type its name in the input line and press <Enter>. The partition table will be exported to disk, with the extension “PTB”. You can use the Import Table function later on with a different physical disk to load the exact same partition structure into the editor, and then write it to the other disk.

If you decide that to completely resize the APT or FAT partitions on the physical disk at a later date, select Disk->Initialize disk from the opening FDISK screen, and then select the disk you want to reinitialize. Be aware that this is a very destructive operation which will result in the loss of all existing information in both the FAT and APT areas of the hard disk.

When editing partitions, FDISK offers both disk and partition level property sheets.

Disk->Properties displays information about the APT area of the physical disk:



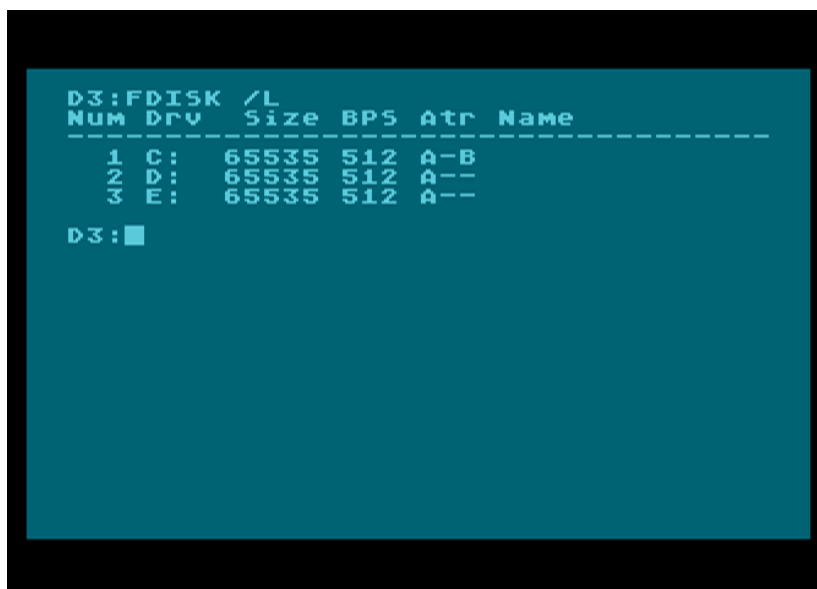
Partition->Properties displays the physical size of a given partition in MB.



In addition to FDISK's role as an interactive partition table editor, you can also use the program to simply list the partition tables on your disk(s). Simply type:

FDISK /L

and FDISK will list the partition tables to the screen, just as they appear on the hard disk (note FDISK's partition table listing does not reflect any changes made to drive letter allocation using MOUNT: see the section on dynamic partition mounting).



By default, FDISK lists the partition tables of all drives present (i.e. both master and slave if you have two drives). To narrow the selection, just follow the /L switch with 1 for master, or 2 for slave.

Support Utilities

The other programs supplied on the support disk are HDDINFO, PARTINFO, DEVINFO and MOUNT.

HDDINFO simply displays geometry and manufacturer information about the physical hard disk. It also shows the exact position (in sectors) of the FAT32 (if present) and APT segments of the disk.

```
D3:HDDINFO
Device name:  GENERIC
      Bus id:  2.0.1.0
Sector size:  512 bytes
      Sectors: 524160

FAT32 Start:  63
FAT32 Size:   260096

      APT Start: 260159
      APT Size:  262144

D3:■
```

DEVINFO (by KMK) displays information about devices connected to the Parallel Bus Interface. The DISKINFO spec used by DEVINFO has been expanded to include SIDE and MyIDE (both of which hook into the LSIO vector and are not PBI devices), so that DEVINFO can report the same information about these interfaces too. The picture below shows typical output from DEVINFO when used with the SIDE driver:

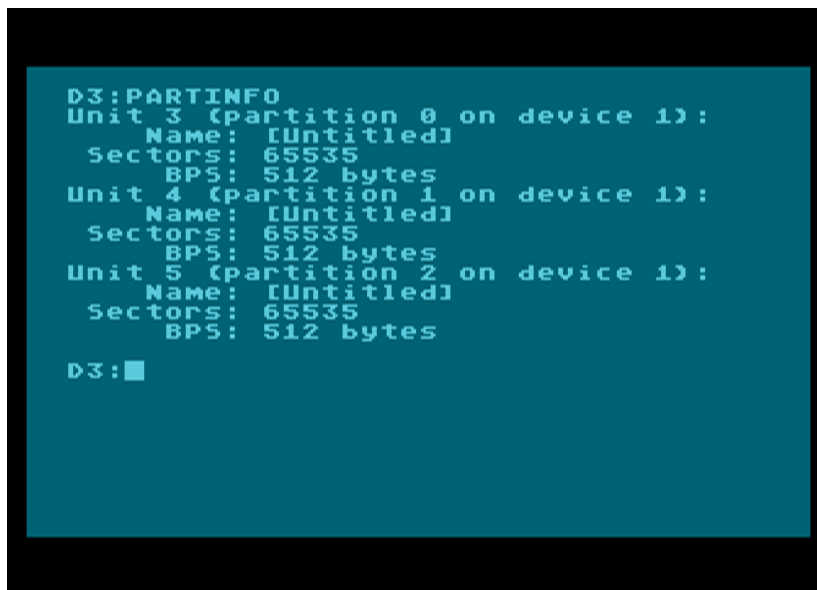
```
D3:DEVINFO

Unit 1:  GENERIC
      Bus id:  2.0.1.0
      Sector:  512 bytes
      Scount:  524160 ($0007FF80)

Unit 2:  No response
Unit 3:  No response
Unit 4:  No response
Unit 5:  No response
Unit 6:  No response
Unit 7:  No response
Unit 8:  No response
Unit 9:  No response
Unit 10: No response
Unit 11: No response
Unit 12: No response
Unit 13: No response
Unit 14: No response
Unit 15: No response

D3:■
```

PARTINFO displays information about mounted partitions, including the drive number, partition name and partition ID for each volume:



```
D3:PARTINFO
Unit 3 (partition 0 on device 1):
  Name: [Untitled]
  Sectors: 65535
  BPS: 512 bytes
Unit 4 (partition 1 on device 1):
  Name: [Untitled]
  Sectors: 65535
  BPS: 512 bytes
Unit 5 (partition 2 on device 1):
  Name: [Untitled]
  Sectors: 65535
  BPS: 512 bytes

D3:■
```

PARTINFO shows drive assignments regardless of the original partition table on disk. If you want to see how the effects of dynamic partition mounting in relation to the partition table on disk, use MOUNT /L (described in the next section).

Dynamic Named Partition Mounting

The second support application is MOUNT. This program permits the dynamic mounting and “ejecting” of partitions during a session, in a similar manner to the MOUNT command found in Linux. Partitions can be addressed either by partition ID or partition name (the partition ID is the same as the sequence number in the left column of the FDISK partition editor).

MOUNT takes several switches and arguments, which you can display by typing:

```
MOUNT /?
```

To mount a partition and attach it to a given drive letter, you can either address the partition by its ID or by its name. To quickly establish the ID of a given partition, you can type:

```
MOUNT /L [device_num]
```

This command will display a list of to all the partitions on a physical disk. “device_num” is an optional number, and should be 1 for the master device, and 2 for the slave device. If no device number is given, the program will first list the partitions on the master device, and will then attempt to list the partitions on the slave device. If there’s no slave device on the system (as is always the case with the SIDE interface), the program will just exit after listing the partitions on the master.

MOUNT’s partition list is different to that of FDISK with the /L switch. While FDISK faithfully lists the drive assignments according to those found in the partition table on disk, MOUNT displays the drive numbers *currently* assigned to partitions, and flags with an asterisk those partitions whose drive number assignments have been changed or removed since the last time the computer was booted or the partition table was written to disk with FDISK.

So how do we mount partitions or change a mounted partition’s drive letter?

First, use FDISK to create at least three partitions on the master disk and give them all unique names. Assign the first partition (ID 1) drive letter C: to use as the boot drive, but leave the rest of the partitions without drive letters. Save the partition table and exit FDISK. Now type

```
MOUNT /L
```

You’ll get a list of the partitions you created, but only one of them will have a drive letter (drive C:, in this case). To mount a partition by ID, include the “/N” switch. Type:

```
MOUNT /N E: 3
```

MOUNT will mount the second partition on drive E. You may include a device number after the partition ID when mounting by ID, so If the desired partition is on the slave device, type:

```
MOUNT /N E: 3 2
```

In either case, you should now be able to type “E:” to log the drive, and format it with FORMAT.

Now we’ll “un-mount” the drive. Type:

MOUNT /U E:

The MOUNT program will remove partition 3 from the partition table in RAM, and drive E will no longer be accessible. Note that at this stage there has been absolutely no permanent link between the drive letter E and partition 3. By using MOUNT to mount the partition, we merely allocated it drive letter E temporarily. Until the partition is permanently allocated a drive letter using FDISK or with MOUNT's "/W" switch (which means "write this change to the partition table"), the partition will not be automatically mounted after a reboot.

If we were to type MOUNT /U C: at this stage, the C drive would also be unmounted, but the difference is, since partition 1 has been permanently allocated drive letter C:, the next time the partition table is re-read into the computer's memory (either with <Shift+Reset> or on the next reboot), drive C will still be online.

OK – back to the command prompt. Now let's mount partition 3 again, but this time by its name:

A screenshot of a DOS command prompt window with a black background and cyan text. The text displayed is: "D3: MOUNT G: BOOT PARTITION", "Partition 1 mounted on drive G", and "D3: █".

```
D3: MOUNT G: BOOT PARTITION
Partition 1 mounted on drive G
D3: █
```

Type:

MOUNT G: [partition name]

(substitute [partition name] with the name you assigned partition two in FDISK, and don't include the square brackets; the name may contain spaces). MOUNT will scan all devices (master and slave if present) for the named partition (note that it's therefore advisable to avoid partitions with the same name on the master and slave drives, since both are always searched for the partition with the specified name). If the named partition is found, it will be mounted on the specified drive letter.

If you now type MOUNT /L, you'll see the partition shows up in the partition list as drive G:, and it has an asterisk next to it. Any partition which was previously assigned as drive G: will now be unmounted (it will have no drive letter), and will also be flagged with an asterisk.

You can also make the changes permanent when mounting a partition. For example:

MOUNT /W /N C: 2

This will cause partition with ID 2 to be permanently mounted on drive C: (the change will be written to the partition table on disk). If there was already a partition mounted on drive C:, it will no longer be attached to any drive letter. The “/W” switch causes a mount operation to be persistent, meaning the partition will be permanently assigned to the drive number you specify, until you change it in FDISK or by using MOUNT again with the “/W” switch.

You should exercise care when using MOUNT. Some points to note:

- Be careful not to un-mount the drive containing your applications and in particular the MOUNT command itself, otherwise you won't be able to get them back. If mount is on the SDX CAR: device (the default arrangement), this risk is avoided.
- The same partition cannot be mounted on more than one drive letter at the same time.
- You must un-mount a partition before you can re-mount it on a different drive letter.
- Mounting and unmounting drives has no effect on the partition table on disk, unless the “/W” switch is included.
- You can currently access up to 127 partitions on a disk.
- Mounting a partition on a drive letter already associated with another partition will cause the other partition to go offline and be replaced by the newly mounted one.
- Mounting a different partition to a drive letter effectively causes SDX to detect a disk change.

Using SDX's batch file facility, one may construct dynamically allocated arrays of partitions if suitable care is exercised during the design stage. In theory, a Compact Flash card of up to 4GB can be completely filled with 32MB partitions, all switched in and out of the filing system on demand.

Benchmarking and Stability Testing

Your APT software distribution may also contain two KMK programs, RWTEST and RWCRC. RWTEST will establish the transfer speed of the hard disk interface, and RWCRC will perform a cyclic redundancy check on a large file in order to ascertain that I/O on the interface is free from errors. A “pass” using RWCRC usually means a stable hard disk interface.

You should obtain the latest versions of RWCRC and RWTEST from the “Other Stuff” page of KMK’s website:

<http://drac030.atari8.info>

Customising the System

If you use the SDX Imaging utility to change the contents of the SDX CAR: device, note that FDISK.COM depends on the presence of the HDDLIB.DLL and UILIB.DLL files, and MOUNT.COM requires HDDLIB.DLL. These two files must be accessible via the SpartaDOS X path in order for the dependant programs to work. If you remove all or part of software suite from CAR:, please ensure that the DLLs are made available to FDISK and MOUNT, otherwise they won’t work.

Credits

Thanks to the following:

- Candle for the SIDE design
- Zaxon for SIDE carts
- KMK and Trub for SIDE SDX ROMs and for patiently answering my questions
- Phaeron for the Altirra emulator with its SIDE support, and for his additional insights into the IDE protocol
- Lewis for beta testing
- Everyone else who helped make this great little project a success

Finally, bugs in software, errors in documentation, and feature requests should be reported to:

fjc@atari8.co.uk

Jonathan Halliday (FJC)

8 November 2011