

APT Partition Editor and Utilities

User's Manual

5th Edition

All software described in this manual is, unless otherwise stated, written by Jonathan Halliday (flashjazzcat).

Contents

Introduction	3
Features and Utilities	3
Tools and Drivers	4
Initialisation and Disk Partitioning	5
General Usage Notes	14
Drivers	15
Rapidisk	16
Support Utilities	18
LSDEV	18
LSDSK.....	19
ATRMNT	20
MOUNT	21
IDEDIAG.....	24
Benchmarking and Stability Testing.....	25
Customising the System.....	25
Credits	26
Feedback	26
Finally	26

Introduction

The Atari Partition Table (APT) is a hard disk partitioning scheme (designed by KMK) intended to provide media compatibility among compliant Atari hard disk host adapters. Any host adapter whose firmware or soft-driver follows the APT specification will be compatible with the APT partition editor (FDISK) and APT utilities (such as those provided on the APT and IDE Plus 2.0 toolkit disks).

Compatible PBI adapters include IDE Plus 2.0, Ultimate 1MB/SIDE, 1088XEL with XEL-CF, Incognito, and IDEa (when used with its APT BIOS). Compatible soft-drivers (which require the use of SpartaDOS X) are available for MyIDE, MyIDE/Flash, MyIDE II, SIDE, SIDE2, XEL-CF and Incognito (the latter when in 'Colleen' mode).

The APT partition editor – FDISK – is compatible with all APT device firmware and drivers and may be used with any disk operating system. Only two tools – ATRMNT and MOUNT – require use of SpartaDOS X, primarily because they require command line arguments.

Features and Utilities

While the IDE Plus 2.0 adapter has its own toolkit (written by KMK) whose tools are generally compatible with all APT implementations, I will focus here not on the IDE Plus 2.0 toolkit disk, but on the APT tools distributed with the U1MB, Incognito, XEL-CF, SIDE, MyIDE, and IDEa APT drivers. The majority of the provided tools may also be used with the IDE Plus 2.0. The APT FDISK partition editor is built into modern 'official' SDX distributions which require it.

A general synopsis of the features of current APT implementations is as follows:

- Support for up to 128 dynamically mountable hard disk partitions (partition limit is imposed by the partition editor, not the APT)
- Support for 15 drives (ATRs and/or partitions)
- Support for cluster sizes of 128, 256 and 512 bytes
- Partition locking and deactivation
- Partition naming
- 'External' (commonly FAT) partitions accessible by both the PC and the Atari
- ATR disk image mounting (Ultimate/SIDE, Incognito, XEL-CF and IDE Plus 2.0 only; IDE Plus uses a proprietary method of ATR mounting)

Tools and Drivers

The following utilities are included on the toolkit disk:

- FDISK.COM
- LSDSK.COM
- LSDEV.COM
- MOUNT.COM
- ATRMNT.COM

The following drivers are also provided:

- SIDE.SYS
- COLLEEN.SYS
- XEL_CF.SYS
- MYIDE.SYS
- MYIDEINT.SYS
- MYIDE2.SYS
- RAPIDISK.SYS

ROM based firmware for the U1MB, Incognito, SIDE/SIDE2 and IDEa may be obtained from <http://atari8.co.uk/firmware/>, where one may also find a detailed user manual describing configuration of the U1MB/Incognito BIOS and use of the U1MB/Incognito/SIDE XEX/ATR loader.

In this document, we shall assume the user has already configured the host adapter and is ready to prepare HDD media for use.

Initialisation and Disk Partitioning

Before using a disk, it must first be initialised. I assume you already have the host adapter attached to the Atari and set up with the appropriate drivers. Power on the machine and boot whichever version of DOS you prefer to use (SpartaDOS X is not a requirement, but it is recommended). If you are using a disk of Compact Flash card without any APT partitioning information on it, the driver may report 'No partitions', or simply respond 'Device does not respond' when attempts are made to access drives other than D1:. This is normal, since we have not yet partitioned the disk.

If you already flashed the appropriate APT-ready SDX build to your hardware, and you are using SDX, you may simply type FDISK at the DOS prompt to run the partition editor, since the application is already on CAR: (user ROM space permitting). Otherwise, mount the APT toolkit ATR on drive 1 and run FDISK by typing its name (under SpartaDOS), or by using your DOS's binary load command. You may also run FDISK from the SIDE loader if you prefer, if using SIDE/Ultimate/XEL-CF/Incognito.

FDISK has an intuitive GUI-style interface which is compatible with the SpartaDOS X S_VBXE.SYS driver (which provides the application with an 80-column display). There's no need to disable the cartridge ROM (i.e. by using the 'X' command under SpartaDOS X) unless you use RC_GR8.SYS to obtain an 80-column display, in which case BASIC or the SDX library must be disabled to make room for the display buffer.

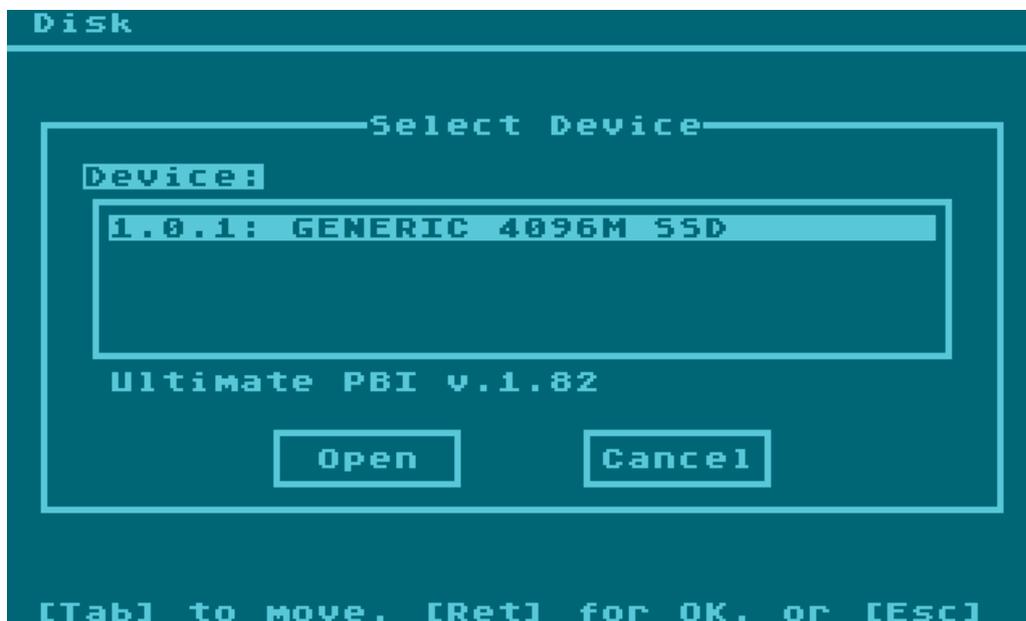
```

APT Partition View Help
-----
Num-Ptn-Start-Size-Drv-BPS-Attr-Partition Name
-----
1 Ext 63 1660865 --- 512 A--- [Unnamed]
2 Ext 1660928 1978368 --- 512 A--- [Unnamed]
3 DOS 3646757 65535 D3: 512 A-BM BOOT
4 DOS 3712293 65535 D4: 512 A--M 130XE ASSEMBLER
5 DOS 3777828 65535 D5: 256 A--- [Unnamed]
6 DOS 3843363 65535 D0: 128 A--- [Unnamed]
7 Ext 4707108 1542177 --- 512 A--- [Unnamed]
8 Ext 6249348 1574307 --- 512 A--- [Unnamed]
9 --- -----
10 --- -----
11 --- -----
12 --- -----
13 --- -----
14 --- -----
15 --- -----
16 --- -----
17 --- -----
-----
Partitions: 8 798147 Free secs
Press [Esc] for menu
    
```

Once FDISK loads, it will display a version notice which will disappear when you open the menus or press a shortcut key. Open the menu with 'Esc' as directed, then press return, and you will see the 'disk' menu:



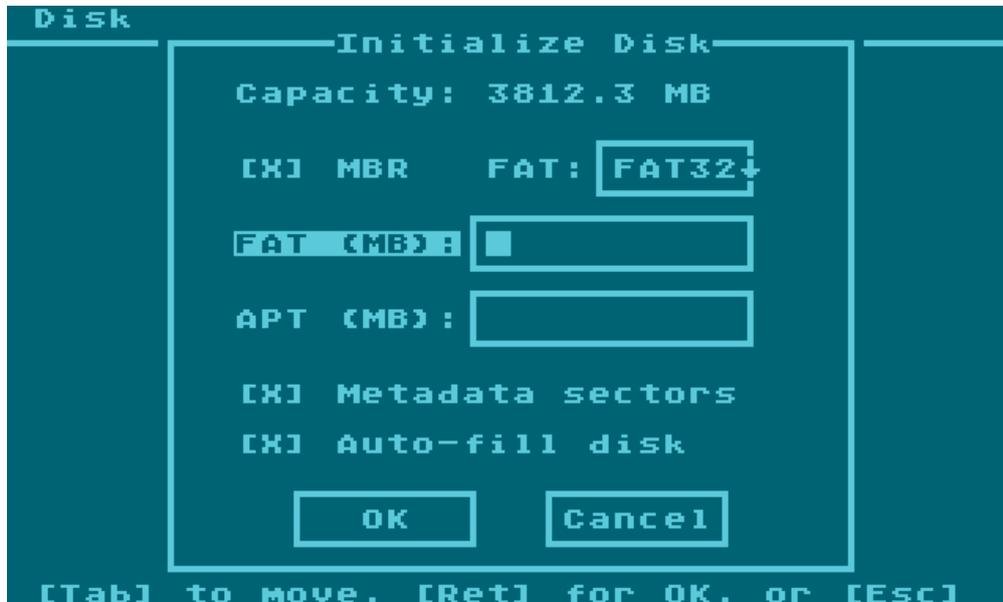
To prepare a new disk or CF card, select 'Initialise...' from the Disk menu. If you already prepared the disk, you can select 'Open...' to open the APT for editing (if no APT is found, you will be directed to the disk initialisation dialog anyway). You will then be presented by the device selector:



The device selector lists every single APT block device on the system, whether a PBI controller or LSIO-attached device driver. It's therefore possible to run different physical devices via different adapters at the same time. Most commonly, though, you will see only one entry here: PBI, controller 0, master disk, which translates to 1.0.1. If you have two physical disks attached to the same adapter, you may also see 1.0.2.

The name of the APT host adapter or driver is also shown, so you can be in absolutely no doubt which device you are working on.

So – having selected the device you wish to initialise by pressing Return, you will may see some warnings if any content is discovered on the disk. If you want to initialise anyway, just ‘OK’ through the alert boxes and you will eventually see the disk initialisation dialog:



Of course, the reported disk geometry will entirely depend on the chosen media. You have several options here, and some decisions to make. Now that FDISK handles ‘external’ partitions, you may wish to have a FAT16 partition at the front of the disk for use with KMK’s FAT drivers for SpartaDOS X. Or you may wish devote half the disk to the SIDE loader (to contain XEX and ATR files), and half the disk for regular APT hard disk partitions. This is the time to decide how you want to arrange the disk.

For the sake of this walk-through, we’ll assume you want the following structure: a FAT16 partition to use with KMK’s FAT driver, and some APT space (which is mandatory) for regular hard disk partitions you can format and use with the DOS of your choice (in this case assumed to be SDX). ‘MBR’ is already checked by default, and this makes sense since the MBR – even when FAT partitions are not required – serves as a protective mechanism to prevent a PC from obliterating the APT. So we’ll leave MBR enabled (to toggle it or any check box, incidentally, press space), and press shift-tab (assuming the ‘FAT (MB)’ text box still has the focus) to move back up to the ‘FAT’ pull-down list. Once that is highlighted, press the down cursor key (with or without control), move the highlight to ‘FAT16’ with the arrow keys, and then press enter to close the list.

Now press TAB to move back to ‘FAT (MB)’. Here’s where you tell FDISK how much disk space you want to use for the FAT16 partition. Notice (providing ‘auto-fill disk’ is enabled) that the size of the APT automatically changes to fill the available free space when you TAB out of the APT (MB) field. You can turn auto-fill off if you want, perhaps if you wish to leave empty space on the disk for additional FATs created later on with some PC-based partitioning tool (since FDISK will only put a single FAT and an APT on the disk for you).

We'll also leave 'Metadata sectors' checked since this is where we can store names for our disk partitions. So, let's move on by pressing Return. We'll then be offered the chance to re-evaluate the chosen partitioning scheme and write it to disk if we are absolutely sure (which we need to be, since once the MBR is written to disk, it will be difficult to recover any information which was previously on the disk).

All being well, the next thing you'll see is the APT editor:



This empty table is where you define the APT partitions which you'll later format with DOS and copy programs and information to. The table has seven columns:

- Num: the partition sequence number
- Ptn: the partition type; commonly DOS, Ext, or F/W (firmware). The default is 'DOS'
- Start: the start position of the partition
- Size: the size of the partition
- Drv: the partition's drive number, if assigned
- BPS: the partition density (128, 256 or 512 bytes)
- Attr: the partition attributes

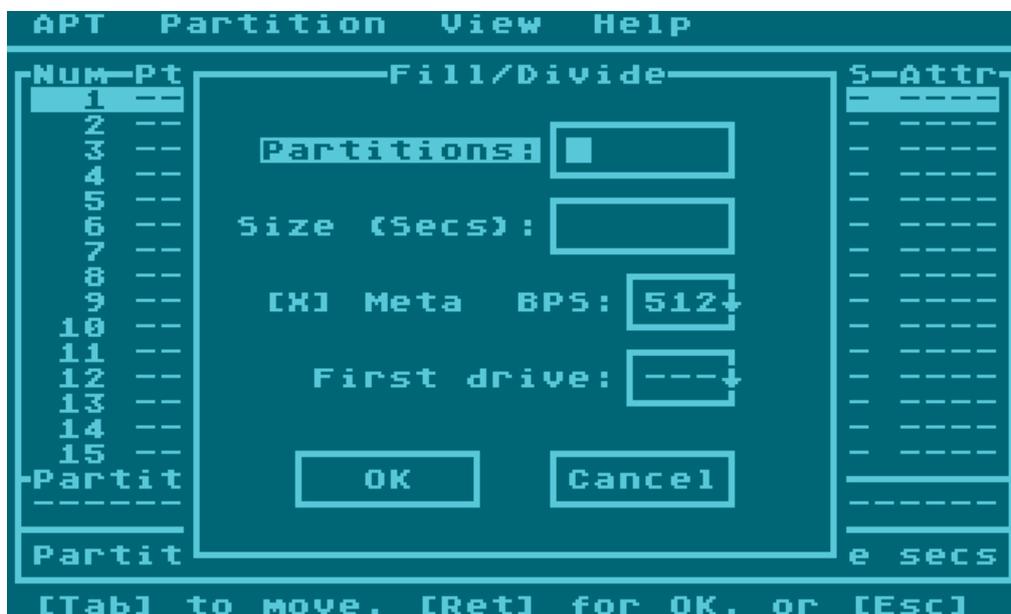
With the highlight on the first entry, simply press Return and type the size in sectors of the first partition you want to create. 65,535 sectors is a common size and the maximum which MyDOS, SpartaDOS and SDX's SPARTA.SYS driver can currently handle. If you prefer to work in megabytes, select View->Megabytes from the drop-down menus and all numerical values will be shown in MB instead of sectors. Note that the 'magic number' is 31.9995MB and no more if you want the partition to be compatible with currently available DOSes.

Once you have entered the partition size and pressed Return, cursor back up the entry and select Partition->Set Drive to give the partition a drive number. It's recommended you start numbering hard disk partitions from D2: or D3: upwards, but it really depends on how you want to configure the

system. Leaving D1: free will allow you to boot from a floppy whenever you need to. In any case, you should be careful to leave at least one drive number unassigned, in case you need access to external SIO storage when the hard disk has every drive number assigned to partitions.

Next, select Partition->Set BPS and change the density of the partition if 512 bytes isn't the size you want. You can choose between 128, 256 and 512 bytes per sector. 256 is the most common size for pre-SDX versions of DOS. Note you can use the shortcut keys listed in the menus to access these options direct from the editor without actually opening the menus at all.

OK – so what if you want eight identical partitions and don't want to type in the details for each one? Well, the latest version of FDISK has a facility which makes this very easy indeed: 'Fill/Divide'. Select Partition->Fill/Divide and type in the details of how you want the program to divide and fill the space in the APT.



If you want eight partitions all of equal size, simply enter '8' in the 'Partitions' field, set 'First drive' to the number you want assigned to the first partition (the rest will follow in series; I'd recommend D3: here), then press Return. FDISK will create eight 512bps partitions of equal size, with drive numbers increasing by one, as near as possible filling all the available space. But if the APT space is large, the partitions may end up more than 65,535 sectors long, which at the moment we don't want.

The solution, then, is to simply specify the size in sectors as well. This will limit the size of the created partitions. Likewise, you can omit 'Partitions' entirely and just type 65535 into the 'Size' field. In that case, FDISK will create as many 65,535 sector partitions as possible to fill the entire APT space. Note that you can also change the sector size here, if for example you want a number of identical 256bps partitions.

Since partitions are commonly created in identical sizes, 'Fill/Divide' is probably the most convenient way to build a partition table. In any case, when you've finished creating the partition table, you should set the boot partition. This will tell the BIOS or drive which partition to boot from when the computer is switched on. Just cursor up to the partition you wish to assign as the boot partition, and

press 'B', or select Partition-Bootable. You'll see the 'B' flag appear next to the partition, under 'Attr' (Attributes).

Note the other flags too: A means 'active', and simply means that the partition will be auto-mounted when the partition table is read when the machine is switched on. Clearing the active flag won't delete the partition, but will make it inaccessible. Removing the drive number effectively has the same effect as clearing the active flag. Next is space for the 'L' flag, which means 'locked'. A locked partition is read-only, and cannot be formatted or written to.

Next we have the 'B' (boot) flag as already discussed, and finally the 'M' flag, which stands for 'Metadata'. Metadata is contained in a special reserved sector, typically right in front of the partition on disk (watch the start sector numbers when editing the partition table when metadata is enabled, and you may notice space being reserved for metadata, as well as for the APT partition table chain itself if many partitions are created). You can turn metadata off so that new partitions do not have metadata by default: do this via APT->Properties (since the setting is too destructive to access direct from the editor). But note that without metadata, partitions cannot be named. In addition, enabling or disabling metadata later on a per-partition basis (which can be done via Partition->Properties) will *move* the partitions and make any data in them invalid. Partitions will also move when partitions ahead of them are added or deleted, so before formatting anything and copying information into the partitions, it's wise to get decisions like metadata and partitions size out of the way at the first instance.

My advice is to leave Metadata globally enabled (the default state).

You should now have a partition table which resembles this:

```

APT Partition View Help
-----
-Num-Ptn-Start-Size-Drv-BPS-Attr-
-----
 1 DOS 3646757 65535 D3: 512 A--M
 2 DOS 3712293 65535 D4: 512 A--M
 3 DOS 3777829 65535 D5: 512 A--M
 4 DOS 3843365 65535 D6: 512 A--M
 5 DOS 3908901 65535 D7: 512 A--M
 6 DOS 3974437 65535 D8: 512 A--M
 7 DOS 4039973 65535 D9: 512 A--M
 8 DOS 4105509 65535 DJ: 512 A--M
 9 ---
10 ---
11 ---
12 ---
13 ---
14 ---
15 ---
-Partition name-
[Unnamed]
-----
Partitions: 8          536001 Free secs
-----
Press [Esc] for menu
    
```

In the example above, we have eight equally sized 512bps partitions, numbered D3: through DJ: (3-10), and D3: set as the boot drive. But what of the FAT partition we created earlier? We want that to be readable in SpartaDOS X...

To get access to the FAT16 partition from SDX, we need to add one last partition: an external partition. Choose Partition-> External, and you will see a list of all the FAT partitions (which is to say, those which belong to the MBR partition table, and are *outside* of the APT area):



We can see the FAT16 partition we created earlier (although it's currently unformatted – we'll do that on the PC later), so hit return and an entry pointing to that FAT will be added to the APT table. The external partition is denoted by a left-pointing wedge at the left side of the entry. All that remains is to assign the external partition a drive number.



Note that, as previously mentioned, we can name partitions. Names are stored in the metadata sectors. To edit a partition's name, just press 'N', or choose Partition->Edit Name. Type the name in the space at the foot of the screen and press enter. The names of partitions are displayed here as you scroll through the list. When using an 80 column display, the partition names are displayed to

partition. There's no point in formatting the FAT partition, since we want a FAT file system there, and we'll do that by using the PC's card reader.

Once you have formatted all the other partitions, you should be able to list the empty root directories in DOS. If everything looks right, remove the disk or CF card from the adapter (if necessary, power down first), and connect to the PC. The PC's operating system should recognize the MBR partition table, and say that the FAT partition requires formatting. Go ahead and format the partition, being sure (since we chose FAT16 when setting it up) that 'FAT' is used (not 'FAT32' in this particular example). Once that's done, copy some files into the newly formatted FAT16 partition on the PC, making sure they have 8.3 names (no long filenames for the moment). Now, remove the disk or card from the PC, reconnect to the Atari, and power-on again.

At the SpartaDOS X prompt, type DIR drive:, where drive is the letter of the external FAT partition you created earlier. If all went well, SDX should report 'Unrecognized disk format'. That's as we expect, since we need to install FATFS.SYS first. If you flashed the provided SDX ROM to your interface, FATFS.SYS is already on CAR:. Otherwise, install FATFS.SYS from the APT toolkit disk, or the SpartaDOS X 4.46 toolkit disk. If you intend to use FAT partitions all the time, you should also add the line 'DEVICE FATFS' to CONFIG.SYS, in which case the driver – if it is not already on CAR: - may reside on the hard disk's boot partition, which must be formatted SDFS.

With the driver installed, you should now have read-only access to the external FAT partition. Well done! The implications of this should be obvious: the FAT partition offers a very convenient way to move files and ATRs in bulk from the PC to the Atari, without the need for specialised tools or file system drivers on the PC side. Moreover, if you copy ATR disk images into the FAT partition (providing you are using Ultimate or Incognito), you can mount these disk images directly, with read/write access using the ATRMNT tool which should already be on CAR:...

With the FAT partition mounted, just type ATRMNT A: ATRNAME to mount an ATR on the specified (unused) drive letter. You can then read from and write to the ATR, just as if it were a regular disk, and when done, copy the changed ATR back to the PC by simply dragging it or copying it from the FAT partition.

Note that IDE Plus 2.0 requires mounted ATRs to reside in an SDFS partition, so if using that adapter, you'll first need to copy the ATRs into an SDFS formatted partition, before mounting them with IDE Plus's proprietary mounting tools (MATR, ATRMNT and MOUNT are intended for use only with Ultimate and Incognito PBI BIOSes).

With Ultimate/Incognito, you may also use MATR to mount ATRs in FAT16 partitions, and this method does not require FATFS. See the documentation for ATRMNT and MATR for further details.

General Usage Notes

Of course, the foregoing guide just covers one typical scenario of disk partitioning. You may instead want to create a FAT32 partition alongside the APT for use with the SIDE loader (or the XEL Loader). You can even create multiple FAT partitions of different types by leaving sufficient space for them when initialising the media with FDISK, then adding more FAT partitions later on using – for example – Windows' Disk Management Console. In that case, it's possible to then add external entries for these FAT partitions just as we did for that single FAT16 partition. Once FAT32 file system drivers are available for SpartaDOS X, it will make sense to have multiple external FAT16 and FAT32 partitions.

As for FDISK, while there are facilities that we have not fully covered here, we have certainly covered basic operation, which is sufficient to get up and running with a usable hard disk. While it is possible to resize and change the properties of already formatted and populated partitions, be aware that since FDISK now insists on there being no unused disk space *between* partitions, removing metadata, inserting external partitions with metadata, deleting partitions, and inserting new ones is likely to invalidate the data on neighbouring partitions. Therefore, you need to bear this in mind when first planning your partitioning scheme.

Drivers

For non-U1MB systems, cartridge-based storage devices incompatible with the U1MB PBI BIOS, and systems running non-PBI-aware operating systems, the following selection of SpartaDOS X mass storage drivers is provided:

Driver	Target hardware	Remarks
SIDE.SYS	SIDE and SIDE2	Supports media hot-swapping with SIDE2
COLLEEN.SYS	Incognito (in 800 'Colleen' mode)	Suppresses itself in XL/XE mode
XEL_CF.SYS	XEL-CF	Supports dual CF cards
MYIDE.SYS	MYIDE and MYIDE Flash	Supports dual ATA devices (older MYIDE)
MYIDEINT.SYS	Internal MYIDE	Supports dual ATA devices
MYIDE2.SYS	MYIDE II	Supports media hot-swapping

These mass storage drivers work only with SpartaDOS X and add an entry to the SDX LSIO table which allows the system to talk to the hard disk without PBI firmware or a patched OS. As a result, the hard disk is only accessible while SDX is present. The problem of the operating system being unaware of the hard disk at boot time (i.e. at the point at which the driver needs to be installed) is neatly solved by the fact that SpartaDOS X is capable of loading drivers directly from its 'CAR:' ROMdisk device. For convenience, the drivers are therefore typically placed on CAR: in ROM and referenced by the default CONFIG.SYS (also on ROM).

The reliance on SDX means that partitions cannot be booted and the cannot be used at all if SDX is disabled. In addition, no provision is made for ATR mounting or booting using the SDX drivers.

Those drivers which support dual devices should have secondary device polling explicitly enabled (if required) via the '/S' switch. In CONFIG.SYS, for example, the following line loads the MYIDE driver with secondary disk polling:

```
DEVICE MYIDE /S
```

The following screenshot depicts a typical SDX boot sequence when using the SIDE.SYS driver:

```

Ultimate clock installed
SIDE Driver v.3.50 (c)2018 FJC
2.0.1: GENERIC 4096M SSD
Partitions: C,D,E,F,G,H,I
Boot drive: C

SpartaDOS X 4.49c 10-07-2018
Copyright (C) 2018 by FTe & DLT

D3:■
    
```

Note the Bus ID of 2, denoting that this is an LSIO device. You can suppress the display of device and partition/boot drive information via the /Q (quiet) switch.

You may use FDISK and any of the supplied tools with the SDX drivers aside from those concerning dynamic mounting (not supported by the soft-drivers) or those which depend on the presence of a PBI-based driver.

The SDX drivers may be found on the toolkit disk and also pre-installed on the CAR: device of customised SDX ROMs.

Rapidisk

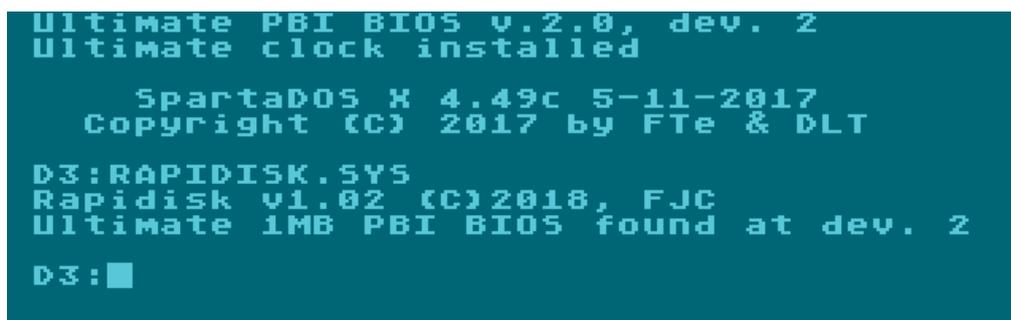
As well as mass-storage drivers, an additional driver (RAPIDISK.SYS) is provided for owners of the Rapidus 65C816 CPU accelerator board. It provides significantly accelerated HDD transfer speeds at 20MHz with the DracOS operating system (or any custom operating system which allows the 65C816 to run in native mode).

The driver is compatible with U1MB/SIDE and the 1088XEL/XEL-CF and would conceivably work with Incognito were it feasible to install Rapidus in an Atari 800. The driver patches the PBI BIOS and moves the transfer code into the last bank of linear 65C816 RAM. Since the 65C816 can execute native code in RAM at a much higher speed than it can execute code in the Atari's ROM (which is tied to 1.79MHz), the patch increases HDD throughput from around 60KB/s to 150-180KB/s.

ATR throughput is greatly improved as well, resulting in RWTEST on a 512bps ATR mounted in a FAT partition achieving speeds at 20MHz similar to those of a native HDD partition at 1.79MHz. Although some functions still run in ROM, the most intensive sections of code run at full speed in RAM.

There are two versions of the Rapidisk driver. The first – RAPIDISK.SYS – requires reinstallation following each system reboot, as per most device drivers. The second version (RAPIDSKR.SYS) is a 'resident' version of the driver which can survive OS reboots providing the DracOS operating system remains present at all times. This reboot-proof version of the driver allows the booting of ATRs at accelerated speed via the SIDE/XEL loader.

Below is a screenshot of RAPIDISK.SYS being loaded from the SDX command line on an U1MB/SIDE2/Rapidus machine. The U1MB PBI BIOS's device ID has been set to 2 to avoid clashing with Rapidus' fixed PBI ID of 0.



```
Ultimate PBI BIOS v.2.0, dev. 2
Ultimate clock installed

SpartaDOS X 4.49c 5-11-2017
Copyright (C) 2017 by FTe & DLT

D3:RAPIDISK.SYS
Rapidisk v1.02 (C)2018, FJC
Ultimate 1MB PBI BIOS found at dev. 2
D3:█
```

APT Partition Editor and Utilities (5th Edition)

Note that the driver will not load unless the operating system is capable of running the 65C816 in native mode. Linear RAM is also a requirement.

Support Utilities

The APT toolkit disk includes a number of support tools which fully integrate with the APT firmware and drivers. Each utility is described below.

LSDEV

APTDEV.COM simply lists all APT devices on every APT controller found on the system, including LSIO devices such as those controlled by the SIDE.SYS driver.

Compatibility:

Any DOS

U1MB/SIDE, Incognito, 1088XEL/XEL-CF, IDE Plus 2.0, SDX soft-drivers

Usage:

LSDEV

Example:

In the example below, LSDEV has been called from the SDX command line, and displays two physical disks (primary and secondary) on the XEL-CF controller. The three digits of the device ID are as follows:

Bus ID

Controller ID

Device ID

Bus ID is '1' in this case, denoting a PBI device. Controller ID is '2', since the PBI ID of the PBI BIOS is currently set to 2 (the possible range being 0-7). Finally, the Device ID is 1 for the primary disk and 2 for the secondary disk.

You'll see these same ID values used to identify physical disks in applications like FDISK, and they offer an unambiguous way of differentiating one controller or disk from another.

```
D3:LSDEV
Device: 1.2.1
Media:  GENERIC 4096M SSD
Cntlr:  U1MB XEL-CF PBI BIOS v.2.01
Size:   4096MB 512bps

Device: 1.2.2
Media:  GENERIC 2048M SSD
Cntlr:  U1MB XEL-CF PBI BIOS v.2.01
Size:   2048MB 512bps
```

LSDSK

LSDSK lists every currently mounted APT HDD partition. Note that partitions may reside on different controllers and physical disks, and a partition's controller may be deduced from its ID.

Compatibility:

Any DOS

U1MB/SIDE, Incognito, 1088XEL/XEL-CF, IDE Plus 2.0, SDX soft-drivers

Usage:

LSDSK

Example:

In the example below, LSDSK has been launched from SDX on a 1088XEL system with a dual-card XEL-CF controller. Logical partitions are spread across two physical disks (primary and secondary), and the components of the partition IDs are as follows:

Bus ID

Controller ID

Device ID

Device Sub-ID

The first three numbers are just as per the output of the LSDEV command and denote the device to which a given partition belongs. In addition, we have at the end the Device Sub-ID, which is simply a partition sequence number unique to that device. You can see that partition 3 of each physical disk is mounted on a logical drive number, but that they have a different unique identifier since their Device IDs differ.

```
D3: LSDSK
D3: HDD: 1.2.1.3      32MB 512bps
    BOOT
D4: HDD: 1.2.1.4      32MB 512bps
    130XE ASSEMBLER
D5: HDD: 1.2.1.1      811MB 512bps
    [Untitled]
D6: HDD: 1.2.1.2      966MB 512bps
    [Untitled]
D7: HDD: 1.2.1.7      500MB 512bps
    LONG NAME TEST PTN
D8: HDD: 1.2.1.8      600MB 512bps
    [Untitled]
D9: HDD: 1.2.1.9      693MB 512bps
    EXTERNAL FAT PARTITION
DJ: HDD: 1.2.2.3      17MB 512bps
    DRIVE C
DK: HDD: 1.2.2.4      17MB 512bps
    DRIVE 9
```

ATRMNT

ATRMNT.COM runs with SpartaDOS X only and is an ATR mounting tool for FAT partitions. It should be used in conjunction with the FATFS.SYS driver.

Compatibility:

SpartaDOS X

U1MB/SIDE, Incognito, 1088XEL/XEL-CF

Usage:

```
ATRMNT [/U] D: [ATRNAME[.ATR]]
```

/U: Unmount drive

Example:

In the following example, the FATFS.SYS driver has been installed in order that SDX can gain access to the 'External' FAT16 partition on logical drive 9 (I). ATRMNT is then called with the desired mount point (drive number) as the first argument, and the path to the ATR as the second argument. The ATR then becomes accessible as logical drive A:.

```
D3:FATFS.SYS
FATFS v.0.87L

D3:ATRMNT A: I:HARDDISK
HARDDISK.ATR mounted on A:

D3:DIR A:

Volume:      HDD
Directory:   MAIN

QCD          COM      4333  23-03-98  23:05
CONFIG       SYS        87  25-01-06  20:14
RWTEST      COM      1762  26-02-18  16:32
SDX          ROM    131072  5-09-10  20:22
```

In the next example, we unmount the previously mounted drive:

```
D3:ATRMNT /U A:
Unmounted A:

D3:■
```

MOUNT

The MOUNT tool allows dynamic mounting and unmounting of APT hard disk partitions via the SpartaDOS X command line.

Compatibility:

SpartaDOS X

U1MB/SIDE, Incognito, 1088XEL/XEL-CF

IDE Plus 2.0 (partition listing only; no mounting)

Usage:

```
MOUNT [dev] drive: part_name
MOUNT /N [dev] drive: part_num
MOUNT /U drive:
MOUNT /L [dev]
```

Dev: [[BusID.]CntrllrID.]DeviceID]

BusID: 1=PBI, 2=LSIO

CntrllrID: 0-7

DeviceID: 1=Primary, 2=Secondary

Switches:

L : List partitions

N : Mount by partition number

U : Unmount specified drive

Example:

Since logical partitions can reside on any one of a number of mass storage controllers and on primary and secondary devices, provision is made by the MOUNT tool to specify the source of a mounted partition via [dev]. The '/L' switch allows listing of all the logical partitions – whether mounted or not – on a physical disk, with their device Sub-IDs in the first column:

```
D3:MOUNT /L
Dev. 1.2.1: GENERIC 4096M SSD
U1MB XEL-CF PBI BIOS v.2.01

 1: D5: 810MB -
 2: D6: 966MB -
 3: D3: 31MB BOOT
 4: D4: 31MB 130XE ASSEMBLER
 5: --- 15MB -
 6: --- 7MB -
 7: D7: 500MB LONG NAME TEST PTN
 8: D8: 600MB -
 9: D9: 693MB EXTERNAL FAT PARTITIO
```

In this example, MOUNT /L displays the partitions on the primary disk (the default).

However, if we wish to view the contents of the secondary disk, we need to specify this:

```
D3:MOUNT /L 1.2.2
Dev. 1.2.2: GENERIC 2048M SSD
U1MB XEL-CF PBI BIOS v.2.01

 1: ---      17MB -
 2: ---      17MB -
 3: DJ:      17MB DRIVE C
 4: DK:      17MB DRIVE 9
 5: ---      17MB DRIVE 0
 6: ---      17MB -
 7: ---      17MB -
```

Here, we have specified the unique physical disk ID after the /L switch.

- 1: Bus ID (PBI)
- 2: Controller ID (2)
- 3: Device ID (2)

If there's only one APT controller on the bus and no soft-drivers (Bus ID 2) present, we can shorten the 'secondary device' notation thus:

```
D3:MOUNT /L 2
Dev. 1.2.2: GENERIC 2048M SSD
U1MB XEL-CF PBI BIOS v.2.01

 1: ---      17MB -
 2: ---      17MB -
 3: DJ:      17MB DRIVE C
 4: DK:      17MB DRIVE 9
```

Here, '2' refers to the device ID, because MOUNT counts incomplete IDs from the *end* of the tuple (i.e. starting at the Device ID). Therefore, if you supplied '0.2', this would denote the secondary device on controller 0, with 'PBI Device' (1) being assumed for the missing first digit.

In practice, assuming you don't use multiple PBI controllers at the same time, it will be sufficient to supply only the Device ID on its own, and even then only on controllers supporting dual physical disks (at the time of writing, XEL-CF and older revision IDE Plus devices).

When mounting partitions by name of Device Sub-ID (partition number), the ID syntax is the same as when using the list switch. You may mount partitions by name where available:

```
D3:MOUNT A: BOOT
Partition 3 mounted on D1:
D3:█
```

If the target partition is unnamed or if you prefer to mount by Sub-ID, you may do so thus:

```
D3:MOUNT /N A: 3
Partition 3 mounted on D1:
D3:█
```

To mount a partition by Sub-ID on a secondary drive, one might use the following method:

```
D3:MOUNT /N 1.2.2 A: 1
Partition 1 Mounted on D1:
D3:■
```

Note, however, that we may omit '1.2.' from the front of the device string as previously discussed if there is only one controller on the bus.

Finally, to unmount a partition, we need simply supply its drive number (mount point):

```
D3:MOUNT /U C:
D3: unmounted
D3:■
```

MOUNT insists that partitions be unmounted before being reassigned to a different logical drive number, and that mounted logical drives are unmounted before another partition is mounted on the same drive number.

Note that dynamic mounts remain active only until the next power-cycle or forced partition table refresh. This means that dynamic partition mounts will be lost in the following circumstances:

- Using FDISK to edit the partition table (on either the primary or secondary disk)
- Performing a disk or partition table refresh in the SIDE or XEL Loader
- Hot-swapping physical disks
- Turning off the machine
- Disabling and re-enabling the PBI HDD in the U1MB or Incognito setup menu

IDEDIAG

Compatibility:

Any DOS

PBI BIOS or SDX driver: U1MB, SIDE, SIDE2, Incognito, XEL-CF, MYIDE, MYIDE II, Internal MYIDE

Usage:

IDEDIAG

The IDEDIAG tool provides a rudimentary hard disk stability test. Its purpose is simply to verify proper low-level communication between the Atari and the hard disk. The tool reports basic information about the primary and secondary devices (if present) and runs a basic verified read test and reports any controller errors.

Since the U1MB and Incognito PBI BIOS will retry HDD transfers if a communication error is detected, IDEDIAG provides a truer indication of whether the computer and the hard disk are communicating with one another properly. Although passing the IDEDIAG test is not an absolute guarantee that a mass storage device will work stably in all situations (in the XEX loader, etc), failing the IDEDIAG test is a reasonable indication that a device will probably encounter issues in general use.

Since IDEDIAG performs low-level writes to the hardware, it requires the presence of an APT PBI BIOS or SDX soft driver in order to configure itself to the host platform. If using SDX drivers, the latest versions must be used.

Example:

This example shows IDEDIAG being run on an U1MB/SIDE system. The secondary device is naturally unresponsive, since only a primary physical disk is present in a SIDE cartridge.

```
D3:IDEDIAG
IDEDIAG v.1.04 by FJC
U1MB PBI v.2.00 found at dev. 0
ATA Reset: ACKed in 0ms ($50)
Disk1 PIO: ACKed in 0ms ($50)

Disk1: (Dev. 1.0.1)
Cylinders: 8322
  Heads: 16
  Sectors: 63
  S/N: 1260141328
  Rev: 1.0
Model: GENERIC 4096M 55D
  Size: 8388608 sectors (4096 MB)

Test passed OK

Disk2: (Dev. 1.0.2)
No response

Test again? (Y/N): █
```

Benchmarking and Stability Testing

As well as the aforementioned IDEDIAG tool, two other useful utilities exist for the purpose of benchmarking disk transfer speeds and verifying the accuracy of data transfer respectively: RWTEST and RWCRC. These tools are written by Drac030 and are available from his website:

<http://drac030.atari8.info>

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 and is a highly recommended test when first setting up your IDE host adapter.

You should obtain the latest versions of RWCRC and RWTEST from the 'Other Stuff' page of KMK's website.

Customising the System

If you wish to edit the default CONFIG.SYS on the CAR: device (necessary when altering the configuration when soft-drivers are used), you may use the SDX Imaging utility available from the SpartaDOS X Upgrade site (<http://sdx.atari8.info/>) for this purpose. Note, you should always include DEVICE [driver name] in CONFIG.SYS, otherwise you will lose access to the hard disk.

The above does not apply to PBI host adapter solutions, such as Ultimate, Incognito, etc.

Credits

Sincere to the following:

- Michael St Pierre for the remarkable 1088XEL project
- Candle for hardware design, support and technical insight
- Zaxon and Lotharek for providing hardware
- KMK and Trub for SDX ROMs, feedback and insights
- Phaeron for the fantastic Altirra emulator and technical knowledge
- Lewis for beta testing the old SIDE drivers
- ProWizard for his tremendous and dedicated assistance in troubleshooting the PBI BIOS and for bringing to my attention bugs missed by everyone else, including myself
- Steve Carden for testing FDISK
- Hias for his assistance and technical insights
- Everyone else who helped

Feedback

If you discover any bugs or experience any issues using the software, or perhaps you simply have some suggestions about improvements which could be made, please contact me via the AtariAge forums, or by email at:

fjc@atari8.co.uk

Finally

If you find the tools and drivers described in this manual useful and wish to offer your appreciation for the hundreds of hours of effort which went into developing them, please feel free to make a donation via www.atari8.co.uk.

Thank you in advance. I hope you find the APT tools useful.

Jonathan Halliday (FJC)

July 2018