

SPARTADOS



SpartaDOS X

The Most Powerful 8-Bit
Disk Operating System

by ICD

For further information on this system contact ICD, Inc. at 815-968-2228 voice or 815-968-2229 BBS.

Note — throughout this manual:

SpartaDOS, SpartaDOS Construction Set, SpartaDOS X, SpartaDOS Tool Kit, FlashBack!, Multi I/O, MIO, P: R: Connection, Printer Connection, UltraSpeed, US Doubler, Action!, MAC/65, BASIC XL, BASIC XE, OS/A+, and DOS XL are trademarks of ICD, Inc.

Atari 130 XE, 800XL, 400/800, 810, 850, 1050, XE Game System, XEGS, AtariWriter, and AtariWriter Plus are trademarks of Atari Corp.

ATR8000 is a trademark of SWP, Inc.

Percom is a trademark of Percom Data Corp.

Axlon RAMPOWER is a trademark of Axlon, Inc.

MSDOS is a trademark of Microsoft Corporation

UNIX is a trademark of AT&T

ARC is a trademark of Systems Enhancement Associates

Published by ICD, Inc.
1220 Rock Street
Rockford, IL 61101-1437
U.S.A.

This manual and the software on the included cartridge are copyright 1987, 1988, 1989 ICD, Inc. All rights reserved. Printed in the United States of America. Reproduction or translation of any part of this work (beyond that permitted by sections 107 and 108 of the United States Copyright Act) without permission of the copyright owner is unlawful.

Notice: In no event will ICD, Inc. be liable for any damages whatsoever (including without limitation, damages for loss of business profits, business interruption, loss of business information or the like) arising out of the use of, interruption in the use of, or inability to use this software, even if ICD, Inc. has been advised of any possibility or likelihood of such damages.

Table of Contents

1 Introduction

2 An Introduction to SpartaDOS

What is DOS?	2-1
SpartaDOS	2-1
The Command Processor	2-2
Getting Started	2-2
Formatting a Disk	2-3
Disk Directory	2-3
Creating Test Files	2-3
Setting the Time and Date	2-5
Parameters	2-6
Copying Files	2-7
Erasing Files	2-7
Wildcards	2-7
Directories	2-8
The Current Directory	2-10
Running Programs	2-11
BASIC, CAR, and X	2-11
Practice	2-12
Building Batch Files	2-12
DOS 2 Equivalents	2-13

3 SpartaDOS X Overview

Filenames	3-1
Extensions	3-2
Wildcard Characters	3-2
Directories	3-3
Device Identifiers	3-5
Default Drive and Directory	3-7
Volume Names	3-7
Disk Format Compatibility	3-8
Using External Cartridges with SpartaDOS X	3-8

4 The Command Processor — Commands

ARC (Archive Files) Command	4-3
ATR (Attributes) Command	4-7
BASIC Command	4-9
BOOT Command	4-12
CAR Command	4-14
CHDIR (Change Directory) Command	4-16
CHKDSK Command	4-18
CHTD (Change Time/Date Stamp) Command	4-19
CHVOL (Change Volume Name) Command	4-19
COLD Command	4-20
COMMAND (The Command Processor)	4-21
COPY Command	4-23
DATE Command	4-26
DIR (Directory) Command & DIRS (Short Directory) Command	4-27
DUMP (Display File in HEX Format) Command	4-29
ERASE Command	4-30
FIND (Find Files) Command	4-31
FORMAT Command	4-32
KEY (Keyboard Buffer) Command	4-37
LOAD Command	4-38
MEM Command	4-39
MENU Program	4-41
File Commands	4-42
Dir Commands	4-43
Xtra Commands	4-44
MKDIR (Make Directory) Command	4-46
PATH (Set Search Directory) Command	4-47
PAUSE Command	4-49
PEEK Command	4-50
POKE Command	4-51
PROMPT (Set System Prompt) Command	4-52
RENAME Command	4-55
RMDIR (Remove Directory) Command	4-56

RS232 (Load RS232 Driver) Command	4-57
SAVE Command	4-58
SET Command	4-59
SWAP (Swap Drives) Command	4-60
TD (Time/Date Display) Command	4-61
TIME Command	4-62
TYPE Command	4-63
UNERASE Command	4-64
VERIFY Command	4-65
X Command	4-66
5 The Command Processor — Advanced Features	
Batch Files	5-1
Default Batch File	5-2
I/O Redirection	5-3
Search Path	5-4
6 Programming with SpartaDOS X	
SpartaDOS X Functions from BASIC	6-1
Notes on the Default Drive	6-1
Accessing the “Kernel” Through CIO	6-2
Open File	6-2
An Example	6-3
Accessing the Raw Directory	6-3
Scan Mode	6-3
Rename File(s) (RENAME)	6-4
Erase File(s) (ERASE)	6-5
Protect File(s) (ATR +P)	6-5
Unprotect File(s) (ATR -P)	6-6
Set File Position - POINT	6-6
Get Current File Position - NOTE	6-8
Get File Length	6-9
Load a Binary File (LOAD)	6-9
Create a Directory (MKDIR)	6-9
Delete a Directory (RMDIR)	6-10
Change Current Directory (CHDIR)	6-10

SpartaDOS X Reference Manual

Set Boot File (BOOT)	6-10
Set Attributes (ATR)	6-11
Format a Disk (FORMAT)	6-12
Get Disk Information (CHKDSK)	6-13
Get Current Directory Path (CHDIR)	6-13
An Example	6-14
SpartaDOS User Accessible Data Table	6-16
An Example	6-18
Vectors Under the OS ROM	6-20
Page Seven “Kernel” Values	6-21
7 Technical Information	
SpartaDOS Disk Format	7-1
Boot Sectors	7-1
Bit Maps	7-3
Sector Maps	7-3
Directory Structure	7-4
Exploring Disks	7-5
8 Configuring Your System	
CONFIG.SYS File	8-1
SPARTA.SYS Driver	8-3
SIO.SYS Driver	8-4
INDUS.SYS Driver	8-5
ATARIDOS.SYS Driver	8-6
RAMDISK.SYS Driver	8-7
CLOCK.SYS Driver	8-8
JIFFY.SYS Driver	8-8
XEP80.SYS Driver	8-9
A Error Messages	
B Command Summary — Alphabetical	
C Command Summary — By Function	
Batch Files	C-1
Directory Commands	C-1
Disk Maintenance Commands	C-2
File Maintenance Commands	C-3

Running ProgramsC-4
Command Processor OptionsC-5
Time/Date SupportC-6
Utilities and Programming AidsC-7
D Miscellaneous Notes	
Using Turbo BASIC XL with SpartaDOS X	D-1
Hardware Configuration	D-1
System Configuration	D-1
Using BASIC XE Extensions	D-2
Using MAC/65 and DDT	D-2
Using AUTORUN.SYS files	D-2
Applications	D-3
Handlers	D-3
BASIC Program Loaders	D-3
Using Batch Files	D-3
Problems with the Atari XF551 and Other Disk Drives	D-5
Formatting Procedures	D-5
Problems	D-5
The XF551	D-5
Other Drives	D-5
E Glossary	
Error Message Summary	

Introduction

Thanks for your purchase of SpartaDOS X. We feel confident that, after you become familiar with SpartaDOS X, you will find it to be the most powerful disk operating system ever produced for your Atari 8-bit computer (or any 8-bit computer, for that matter).

If this is your first experience with any version of SpartaDOS, it would be wise to follow the step-by-step examples presented in **Chapter 2**. This will help you to learn how SpartaDOS operates and how you can take advantage of its power.

Chapter 3 describes the general operation of SpartaDOS X and the ways in which it differs from previous versions of SpartaDOS. Every user should read this chapter.

Chapter 4 describes the SpartaDOS X command set in detail, providing complete information for each command. Until you become familiar with this command set, you will probably be referring to this chapter often.

Chapter 5 discusses advanced features such as batch files, I/O redirection, and search paths. This information is not required to use SpartaDOS X, but can help you take full advantage of these features to save time, perform complex tasks easily, and configure your system for optimum use.

Chapter 6 covers programming, including simple BASIC statements for novice programmers and detailed machine language access to the inner workings of SpartaDOS X for advanced programmers. Examples are given in BASIC and assembly language to aid the programmer in integrating these concepts into his or her own programs.

Chapter 7 follows with a technical and detailed look at SpartaDOS X disk and directory structure. This chapter will probably not be of interest to most users but was included for those interested in creating complex programs.

SpartaDOS X Reference Manual

Chapter 8 provides information for tailoring the configuration of SpartaDOS X to your system. In this way you can take full advantage of your computer's modifications (such as extended memory) and peripherals to provide maximum flexibility and control.

The appendices cover such topics as error messages, command summaries, and common problems and solutions.

SpartaDOS X is by far the most complex disk operating system available for the 8-bit Atari. The very things that make SpartaDOS X so powerful may also make it more difficult to learn than other disk operating systems. If you should have any problems at any time with SpartaDOS X or any other ICD product, don't hesitate to call us (voice 815-968-2228, BBS 815-968-2229) or contact us through electronics mail on GENie (ICDINC) or CompuServe (76004,1600). We appreciate all the input we can get and may have suggestions on how to solve your problem.

A note about incompatibilities: there are some programs that just will not work with SpartaDOS X (or any other DOS for that matter). Some programs are protected or have a DOS built into them. SpartaDOS X is more compatible with other programs than any previous version of SpartaDOS, but there will always be a handful of programs that will not work.

An Introduction to SpartaDOS

This chapter is specifically for those of you who have no prior experience with SpartaDOS and may be somewhat confused by all of this. If you are a user of SpartaDOS 2.x or 3.x, you may wish to simply review this chapter and move on to chapter 3, which outlines the new features found in SpartaDOS X.

What is DOS? DOS stands for Disk Operating System. The primary purpose of DOS is to allow the computer to communicate with one or more disk drives. In practice, a DOS provides many more useful features.

SpartaDOS SpartaDOS X serves these purposes and more. It may take a little longer to learn than other types of DOS for the 8-bit Atari computer, but it will provide such a degree of power and control over the computer that the learning period will be well worth it. This chapter contains, in a tutorial format, several examples of elementary SpartaDOS operation. If you read through the chapter and perform all of the examples, taking time as necessary to experiment, you should be well on your way to becoming a SpartaDOS power user.

The Command Processor SpartaDOS differs from Atari DOS 2 and clones in many ways, but the most apparent is the *user interface*, the way in which you communicate with DOS and DOS communicates with you. Atari DOS and most others are menu driven; all options available are printed on the screen and may be selected with a single letter. SpartaDOS uses a command processor (CP); at a prompt, the full command is typed. Those of you familiar with MSDOS on IBM personal computers and clones, CP/M, UNIX, and many operating systems on other computers will recognize the CP interface. Each of these user interfaces has strong and weak points, but most who take the time to become familiar with the operation of SpartaDOS find its CP interface to be much more powerful and flexible. A menu is also included to make file maintenance, especially with multiple files, very quick and easy.

Getting Started To get started, insert the SpartaDOS X cartridge into the cartridge slot (the left slot on an 800) and turn the computer on. If you do not have an R-Time 8 clock cartridge in the cartridge slot (either slot with the 800), you will see the following message:

```
R-Time 8 not present
```

If you have an 800 with 48K or a 1200XL, 600XL, 800XL, 65XE, or XEGS with 64K of RAM, you will see the following message:

```
RAMDISK not installed!  
No extended RAM available!
```

If you have a stock 130XE (128K of RAM) or a 1200XL, 800XL, 65XE, or XEGS with 128K of RAM, you will see this message:

```
RAMDISK not installed!  
4 banks reserved for XE programs.
```

You will then get a message giving the version number and date of your SpartaDOS X cartridge and a copyright notice,

```
SpartaDOS X 4.xx mm-dd-yy  
Copyright (c) 1988 by ICD, Inc.
```


(with the xx being the current revision number and the mm-dd-yy being the current revision date) followed by a prompt:

```
D1:
```

Formatting a Disk

Before you can begin to explore the command processor, you need to have a floppy disk with which to experiment. Place a new disk in drive #1 and type

```
D1:FORMAT
```

and press the RETURN key. You do not need to type the D1:. Throughout *this* chapter, you will only need to type the text that is in **bold face**. You should press return after each line typed unless instructed otherwise.

After typing FORMAT and pressing the RETURN key, you will see the ICD Disk Formatter menu. Press the 1 key to select drive #1. Press the V key, type TESTDISK, and press the RETURN key. This sets the *volume* name of the disk to "TESTDISK". Make sure you have a new disk in drive #1, press the F key, and press the RETURN key to format the disk. The other options on this menu are described in chapter 4 under the FORMAT command. You can just ignore them for this exercise.

Disk Directory

After the format is completed, press the ESC (escape) key to return to the D1: prompt. Type

```
D1:DIR
```

You should now see something similar to

```
Volume:      TESTDISK  
Directory:  MAIN
```

```
713 FREE SECTORS
```

The actually free sector count may vary depending upon the type of disk drive you are using.

Creating Test Files

The disk has no files on it now. To be able to experiment with commands, we need some files on the disk. If you have an XL or XE computer (except for the 1200 XL) type

SpartaDOS X Reference Manual

D1: **BASIC**

If you have an 800 or 1200XL make sure that you have the Atari BASIC cartridge installed in the top of the SpartaDOS X cartridge. If you do not, turn the computer off, insert the BASIC cartridge into the SpartaDOS X cartridge, and turn the computer on again. With an 800, it is necessary to "fool" the computer into thinking the cartridge door is closed by holding the door switch down. This can be accomplished by sticking a cotton swab, pencil eraser, or other small item into the hole located on the front right edge of the cartridge area. Type

D1: **CAR**

Regardless of computer type, you should now see the familiar BASIC prompt:

READY

Type in this short BASIC program:

```
10 OPEN#1,8,0,"D1:TEST.DAT"  
20 FOR X=0 TO 255  
30 PUT#1,X  
40 NEXT X  
50 CLOSE#1
```

Type in

```
SAVE "D1:TEST.BAS"
```

to save the program and then type

```
RUN
```

to execute it. At the READY prompt return to DOS by typing

```
DOS
```

If you get this error message:

```
Disk Error while saving memory!  
ESC to Abort - RETURN to enter DOS
```

don't panic, just hit the RETURN key. This simply means that you do not have enough memory in your computer to set up a default RAMDISK. The MEM.SAV feature of SpartaDOS X attempted to save the program to this RAMDISK and could not. It takes 128K on an 800 and *over* 128K on an XL or XE computer to establish a default RAMDISK.

Now type

```
D1:DIR
```

You will now see

```
Volume:      TESTDISK
Directory:   MAIN

TEST      BAS      147   7-25-88  10:49a
TEST      DAT      256   7-25-88  10:49a
      709 FREE SECTORS
```

As you can see, this directory format is different from that produced by any other DOS. The first one to eight characters (four in this case) are the *filename*. The next zero to three are the *extension*. The number to the right of the extension is the length of the file in *bytes*, not sectors. This is followed by the date and time. Your date, time, and free sector count may vary. "TEST.BAS" is the BASIC program you just typed in, and "TEST.DAT" is the data file it created.

Setting the Time and Date

If you do not have an R-Time 8 cartridge you will need to manually set the time and date. This will allow you to keep track of when your files were created and which is the latest version of a particular file. To do this, type

```
D1:DATE
```

You will see something similar to

```
Current date is 7-25-88
Enter new date:
```

Enter the date in the format "mm-dd-yy" and press the RETURN key. Now type

```
D1:TIME
```


You will see

```
Current time is 10:55:32
Enter new time:
```

Again, the actual time and date displayed may vary from the examples. Now enter the new time and press RETURN. The time should be in the format “hh:mm:ss” and based on a 24 hour clock (for example, 5:30 PM would be 17:30:00). The time and date are set. Now type

```
D1:CHTD TEST.BAS
```

This will *CH*ange the *T*ime and *D*ate of “TEST.BAS” to the current time and date. Now type

```
D1:DIR
```

You will see that the file “TEST.BAS” now has the current time and date. Now type

```
D1:TD ON
```

This turns the *T*ime and *D*ate display *ON* at the top of the screen. This display may be disabled by typing

```
D1:TD OFF
```

Parameters Many commands require *parameters*. In the above example for the command CHTD, the filename “TEST.BAS” part was a parameter. With TD, “ON” and “OFF” were parameters. A parameter is additional information passed to the command by typing it after the command on the same line. Many commands use more than one parameter. Parameters should be separated from the command and from each other by spaces (although commas are allowed with SpartaDOS X and certain commands). Some commands, such as TIME and DATE, use no parameters. Some parameters are required, while others are optional. Often default values are assumed if no parameters are provided. Since this information varies from command to command, consult chapter 4 for the various required and optional parameters for each command.

Copying Files The command to copy files is COPY. This can be used to copy a file from one disk or directory to another or to copy a file to the same disk with another name or path. Type

```
D1: COPY TEST.BAS MAKEDAT.BAS
```

and list the directory (with the DIR command). You will note that the file "MAKEDAT.BAS" has the same length, time, and date as the file "TEST.BAS", because it is just another copy of the same file. To copy a file from one disk to another with only one drive you must use the MENU command (see chapter 4).

Erasing Files Erasing a file removes it from the disk. While it is possible in some cases to recover a file that has been erased, it is a good idea to be very careful when erasing files. Type

```
D1: ERASE TEST.BAS
```

and list the directory. The file TEST.BAS is gone.

Wildcards Most SpartaDOS X commands allow you to select more than one file by using *wildcards* in place of a character or characters. In poker, a wild card can be substituted for any other card. Wildcards in SpartaDOS perform a similar function.

There are two wildcards used in SpartaDOS and most DOS types. These are the '?' and '*' characters. The '?' represents any character in the given position. The '*' represents any or no character in the given position and in the rest of the positions of the filename or extension. In practice, the '*' is used often, while the '?' is rarely used.

To properly explore wildcards you will need some more files on your disk. Type in the following lines:

```
D1: COPY TEST.DAT ABCDE.DAT
D1: COPY TEST.DAT ABZDE.DAT
D1: COPY TEST.DAT ABCRAIG.DAT
D1: COPY TEST.DAT TEST.DOG
D1: COPY TEST.DAT TEST.DZT
D1: COPY TEST.DAT ABCDE.ICD
```

Earlier it was mentioned that some commands have default parameters if none are provided. DIR is one such command. The proper syntax is

```
DIR [fname.ext]
```

The default “fname.ext” (the one that is assumed if none is entered) is “*.*”, meaning a file that has any or no characters for the name and any or no characters as the extension. Obviously this would include any files, so a complete directory is displayed.

What this means is that you can add a file name and extension to the DIR command to get a partial listing the directory. Type

```
D1:DIR TEST.DAT
```

Since only one directory entry matches that name, it is the only one listed. Now try

```
D1:DIR *.DAT
```

Several files are listed, but only those with an extension of “DAT”. Now try

```
D1:DIR A*.D*
```

This shows only those files whose name starts with “A” and whose extension starts with “D”. Now try

```
D1:DIR AB?DE.DAT
```

The “?” means any character, so “ABCDE.DAT” and “ABZDE.DAT” will be selected. Play around with DIR and different file masks (like “ADC*.D?T” and anything else you can think of) until you feel comfortable with the concept of wildcards.

Wildcards should be used with care with the ERASE and RENAME commands, since they can be used to easily erase or misname multiple files.

Directories What you have seen so far when listing the disk directory is the *main* directory (it even says so at the top). SpartaDOS allows you to add other directories to the disk.

Picture the disk as a filing cabinet. When you want to access a file, you (or SpartaDOS) have to check the whole drawer until the file requested is found. This is no problem if there are not many files in the drawer. Just think how time consuming it would be, though, to have to search through a stack of 100 files every time you wanted a file from the drawer. There are also times when it would be useful to group similar files together, such as keeping all of your paint program picture files together.

Subdirectories can be thought of as folders in the filing cabinet. If you took all of your picture files and put them in a folder labeled "PICTURES", then you would only have to search the drawer until you found the "PICTURES" folder, then search it for the desired picture file. Similarly, you could place all of your BASIC programs in a folder named "BASIC", your text files in a folder named "TEXT", and so on. Then, instead of holding a large stack of loose files, your filing cabinet would hold a well organized collection of folders. Searching through them for the proper one would be much easier than checking every file.

Subdirectories may also be placed within subdirectories, as deeply as you desire. Within the "PICTURES" subdirectory above you could create a "WILDLIFE" directory and a "CARTOONS" directory, for example, and place the appropriate files in these.

Type in

```
D1:MKDIR TESTS
```

This means *MaKe DIRectory* TESTS. Now list the directory. You will see an entry that looks like

```
TESTS          <DIR>  11-11-88 11:59:59p
```

As always, your actual time and date will be different. You have just created a subdirectory, or "folder", named "TESTS". It is, however, empty. To verify this, type

```
D1:DIR TESTS>
```

The '>' character signifies that the preceding name was a directory, not a file (see the previous section of wildcards). You should see an empty directory that looks just like an empty disk except that the

```
Directory: MAIN
```

has now been replaced by

```
Directory: TESTS
```

to show you that it is the directory TESTS, not the main directory. The free sector count will also be less than that of an empty disk.

COPY is another command that allows wildcards. The following line will copy all of the files ending in ".DAT" from the main directory to the subdirectory TESTS:

```
D1: COPY *.DAT TESTS>*. *
```

Now get rid of the .DAT files in the main directory:

```
D1: ERASE *.DAT
```

List the directory to assure yourself that the ".DAT" files are gone. List the directory of the TESTS subdirectory as you did before. You have moved the .DAT files from the main "stack" to the TESTS "folder".

The Current Directory

The current directory is the one that is assumed when none is specified. In all of the examples so far the current directory has been the main directory. This can be changed with a simple command. Type

```
D1: CHDIR TESTS
```

This means *CH*ange *DI*rectory. Now type

```
D1: DIR
```

You will get the directory of TESTS, not the main directory. Now whenever this disk is referenced without a directory mentioned, you will get the directory TESTS. Create another directory and change the current directory to it:

```
D1: MKDIR ANOTHER
```

```
D1:CHDIR ANOTHER
D1:CHDIR
```

The last CHDIR with no arguments will show the path from the main directory to the current directory. In this case you will see

```
>TESTS>ANOTHER
```

This means that you are in the directory ANOTHER which is in the directory TESTS which is in the main directory.

More information on subdirectories is provided in chapter 3. While subdirectories are invaluable for owners of large capacity floppy drives and hard drives, they are generally not needed with standard floppy drives unless a large number of very small files (such as fonts) are on a disk.

Running Programs

To run binary files from SpartaDOS X you just type in the name of the file. For example, to run a program named BALLSONG.OBJ, type in

```
D1:BALLSONG.OBJ
```

If no extension is given, .COM is assumed. To run a program without an extension, then, it is necessary to follow the file name with a period. For example, if the name of the file was DEMO, you would have to type

```
D1:DEMO.
```

If you left off the period, SpartaDOS would try to run a program named DEMO.COM.

BASIC, CAR, and X

As demonstrated earlier, to enter internal BASIC in the 600XL, 800XL, 65XE, 130XE, or the XEGS type

```
D1:BASIC
```

and to enter an external cartridge (such as Action!, MAC/65, BASIC XL, BASIC XE, etc.) type

```
D1:CAR
```


It is never necessary with SpartaDOS X to hold down the OPTION key to disable internal BASIC while booting or to remove the external language cartridge. However, for programs that would ordinarily require the removal of these cartridges, it is necessary to use the X command. For example, DISKRX, from the SpartaDOS Toolkit, will not run with any cartridges installed. To run it from SpartaDOS X, it would be necessary to type

```
D1: X DISKRX
```

This will probably be necessary for most of your large binary load files, since few have been written to avoid cartridge memory.

Building Batch Files

A “Batch File” is simply a file containing a list of commands, one on each line, that you wish the computer to perform automatically. Each line contains the command exactly as you would type it in. A batch file can have any legal file name, but the extension “.BAT” is assumed. Batch files are executed by typing a hyphen followed immediately (no space) by the filename. For example,

```
D1: -TEST
```

would execute the batch file “TEST.BAT”, while

```
D1: -DO_IT.TXT
```

would execute the batch file “DO_IT.TXT”.

SpartaDOS X will automatically execute a batch file on D1: called “AUTOEXEC.BAT” when booted if one exists. This allows you to have several commands executed every time you boot the computer. For example, suppose that you do not have an R-Time 8 cartridge, but you still wish to install the time/date line at the top of the screen and set the proper values. Type the following line:

```
D1: COPY CON: D1: AUTOEXEC.BAT
```

What this will do is copy from the CON: device (the screen editor) to a file named “AUTOEXEC.BAT”. The cursor will move to the start of the next line on the screen. Type in the following, ending each line with a RETURN:

TD ON
TIME
DATE

After you have entered the last line (and followed it with a RETURN), press CONTROL+3 (also referred to as ^3). To do this, press and hold the 'CONTROL' key. While holding down the control key, press the '3' key. This will signal the computer that the end of the CON: "file" has been reached. List the directory. You should now have a file called "AUTOEXEC.BAT" on D1:. Turn the computer off, wait a few seconds, then turn it on again. You should see the commands being executed from the batch file. Enter the TIME and DATE as before.

Batch files are one of the most useful features of SpartaDOS X. You can create them as above or with any word processor or editor that will save a file as straight text (without formatting commands).

Practice Now that you have a basic understanding of the operation of SpartaDOS and the command processor, the best thing to do is play around with the commands covered in this chapter and the rest of the commands found in chapter 4. With practice, you will soon have the most commonly used commands memorized and will feel very comfortable with the CP interface. In fact, the next time you boot with Atari DOS you may feel somewhat restricted by the menu!

SpartaDOS X Reference Manual

DOS 2 Equivalents	The following is a list of the commands from the Atari DOS 2.0s menu and their equivalents in SpartaDOS X:
A - DISK DIRECTORY	DIR and DIRS
B - RUN CARTRIDGE	BASIC for internal BASIC in XL/XE computers, CAR for an external cartridge
C - COPY FILE	COPY
D - DELETE FILE	ERASE , DELETE , or DEL
E - RENAME FILE	RENAME , REN
F - LOCK FILE	ATR +P
G - UNLOCK FILE	ATR -P
H - WRITE DOS	not needed
I - FORMAT DISK	FORMAT
J - DUPLICATE DISK	COPY , MENU
K - BINARY SAVE	SAVE
L - BINARY LOAD	program name, LOAD
M - RUN AT ADDRESS	no equivalent
N - CREATE MEM.SAV	SET CAR and SET BASIC
O - DUPLICATE FILE	MENU

SpartaDOS X contains numerous commands that have no equivalent in Atari DOS 2.0s. SpartaDOS X also supports floppy drives of all sizes in single, dual, and double density, hard drives, RAMDISKS, time/date stamping, subdirectories, and more.

SpartaDOS X Overview

This chapter is an overview of SpartaDOS X filename and pathname conventions, device identifiers, and general usage. It is assumed in this chapter that you have a working knowledge of the command processor. Please refer to chapter 2 if you get confused.

You will find that many of the features described here are new to SpartaDOS X — in many ways SpartaDOS X is even more like MSDOS than prior versions of SpartaDOS were. As a simple example, SpartaDOS X allows drive identifiers like “A:” and pathnames like “\DOS\SUB2\MYPROG.BAS”. Also, many of the new commands pertain to features of MSDOS.

Filenames The basic form of the filename is identical to SpartaDOS 3.2 — it consists of a name and an optional extension separated by a period. Legal characters are as follows

The letters 'A' to 'Z' — lowercase letters are converted to uppercase letters.

The digits '0' to '9' — filenames *may* start with a digit.

The underscore character ('_')

Throughout this manual, we use “fname.ext” to represent a filename. The “fname” portion may be up 8 characters in length, and the “ext” portion may from 0 to 3 characters in length and is optional.

Filename Extensions It is important to develop a standard for naming files. The most common method is to reserve specific extensions for certain types of files. The following list contains some of the most commonly used extensions and their corresponding file types.

.ACT	An Action! source program
.ARC	A compressed archive of one or more files
.ASM	An ASCII machine language source file
.BAS	A BASIC SAVEd program
.BAT	A SpartaDOS batch file
.BXE	A BASIC XE SAVEd program
.COM	A SpartaDOS external command or binary program
.DAT	A data file
.DOS	A disk-based version SpartaDOS
.LST	A LISTed BASIC program
.M65	A MAC/65 SAVEd machine language source file
.OBJ	A binary object code file
.PRN	A listing to be printed
.SYS	A SpartaDOS system file or driver
.TXT	An ASCII or ATASCII text file

In some cases the extensions will be assumed by the command processor or application. For example, “.COM” is assumed for command programs, “.BAT” for batch files, “.SYS” for drivers, and “.ARC” for archives.

Wild Card Characters Two wildcard characters (‘*’ and ‘?’) can be used to take the place of characters in a filename in order to represent a range of filenames. The question mark (‘?’) is a “don’t care” character — it will match any character in its position. For example

```
DIR AB?DE.XYZ
```

lists all directory entries on the default drive with filenames that have five characters, begin with “AB”, have any next character, have “DE” for the next two characters, and have an extension of “XYZ”, such as

```
ABCDE.XYZ  
AB_DE.XYZ  
ABZDE.XYZ
```

and similar file names in the current directory.

The asterisk ('*') in a filename or extension indicates that any or no character can occupy that position and all remaining positions in the filename or extension. For example,

```
DIR AB*.XYZ
```

lists all directory entries on the default drive with filenames that begin with "AB" and have extensions of "XYZ", such as

```
ABCDE.XYZ  
ABCRAIG.XYZ  
AB.XYZ
```

and similar entries. It is important to note that any characters *after* the asterisk in either the filename or extension will be ignored, so that

```
DIR AB*DE.X*Z
```

will list the following directory entries, assuming that they exist:

```
ABCDE.XYZ  
ABCRAIG.XXX  
AB.X
```

Directories The disk is broken up into directories, each of which may contain up to 1,423 entries (earlier versions of SpartaDOS have a limitation of 126 entries — in fact those versions will not read SpartaDOS X directories beyond the 126th entry!) The root directory is named "MAIN" and other directories (which are called subdirectories) can be created under "MAIN". (See the MKDIR command.)

Note: While SpartaDOS X will support up to 1,423 entries in each directory, it is recommended that you try to avoid having more than 200 or so. The size of the directory must increase to allow additional files, and, once increased, will never decrease. A directory holding 1,423 entries would be 32K in size. Large directories will slow down disk access considerably, especially when opening new files.

When you display a directory, subdirectories will appear in the listing with a "<DIR>" in the file size field. Subdirectories may be nested with no limits other than disk space and practicality.

Pathnames Since SpartaDOS can have more than one directory on each disk, it uses a path to describe the route from one directory to another. The characters '>' or '^' are used as directory name separators. If used at the beginning of a path, they tell SpartaDOS to begin at the root directory (MAIN). Also, if one or more '<' characters (or '..\' strings) begin a pathname, SpartaDOS will back up one level toward the root directory for each occurrence. Here are some sample pathnames

```
>DOS>CHTD.COM
\DOS\CHTD.COM
TEMP>JUNK>TEST.DAT
<EXPRESS>EXPRESS
..\EXPRESS\EXPRESS
```

The first two are equivalent — from any directory they both access the file “CHTD.COM” in the “DOS” subdirectory of “MAIN”. The third example accesses the file “TEST.DAT” in the subdirectory “JUNK” which is in the subdirectory “TEMP” which is in the current directory. The fourth and fifth are equivalent —they both access the file “EXPRESS” in the subdirectory “EXPRESS” which is in the parent directory of the current directory.

Note: Since '<<' is used by SpartaDOS X for input redirect, it is necessary to precede two or more '<' characters with a colon when they are intended as directory specifiers. The colon simply means the default drive and directory, but it keeps the '<<' off the front of the command line argument and prevents it from being interpreted as a redirect command. For example,

```
DIR <<PICTURES>
```

does *not* (as it did in previous versions of SpartaDOS) give the directory of the pictures subdirectory found in the directory two toward the root. Use instead

```
DIR :<<PICTURES>
or
DIR ..\..\PICTURES\
```

The longest pathname SpartaDOS X allows is 63 characters. This has no effect on the maximum number of levels of nesting, but does impose a practical limit of about 8 levels.

Command Length The maximum length of a line that will be accepted at the command line is 63 characters. There is no warning when this limit is exceeded. The additional characters will simply be ignored. This 63 character limit includes the command name itself but not the prompt.

Device Identifiers As you may recall, in earlier versions of SpartaDOS the device identifiers were the same as you would use through BASIC or anywhere else in the system — they were all based on the CIO device table. SpartaDOS X is much more flexible. We added a second CIO type entry point into the system (the “Kernel”) and made up device names for the same resources as the standard CIO. For example the standard I/O device (the “editor”) in the CIO is referred to as “E:”, but in the SpartaDOS X “kernel”, it is referred to as “CON:”.

There are a number of reasons we chose to deviate from the standard “Atari” way of doing things:

- Ultimately, we wanted SpartaDOS X to feel exactly like an MSDOS machine. It is a standard — for better or worse. We feel SpartaDOS X is a good teaching tool to this end.
- Since we were making the X cartridge ROM a file oriented device (much like a disk), we needed a device specifier for it.
- We had to have the “kernel” with a device system entirely independent of the CIO IOCB tables anyway. There are simply too many technical problems in using the CIO when you need a high degree of flexibility.
- Because we wanted to refer to drives by either a letter or number, the letters would have conflicted with already existing devices.

(And there are many more reasons — the above are just a few.)

So, here is what we have come up with. Through the command processor, the devices are as follows:

- A: ... I: The letters 'A' through 'I' represent the drives 1 through 9 when used without a device name (a three letter name) in front — the device "DSK" is *always* assumed if none is specified. Lower case is treated as if it were upper case — always!
- 1: ... 9: The numbers '1' through '9' represent the drives 1 through 9 as above — a "2:" is *absolutely identical to a "B:"*!
- Dx: A single 'D' (or 'd') preceding a letter or number is simply ignored. (Thus "D2:" or "Db:" means drive 2 as always.)
- DSKx: "DSK" is the official device identifier for your drives — since it is assumed, you need never type it.
- D: No, this is not the default drive or drive 1 — *it is drive 4*
: Since there is no drive letter, this is the default drive.
- CAR: "CAR:" is the X cartridge "ROM disk" — you may load files from it or do directories of it, but of course you may not save or write to it.
- CON: "CON:" is the standard I/O device (in prior SpartaDOS versions, this was called "E:")
- PRN: "PRN:" is the printer (you may follow the "PRN" with a printer number 1-4 or A-D).
- COM: "COM:" is the RS232 port (again, you may follow "COM" with a port number). There is no default driver for this device.

Now you may ask, "How do I access these devices through the CIO in BASIC?" Well, this is simple — precede the device or drive number by a 'D'. Here a few examples to emphasize the point.

```
OPEN #1, 4, 0, "D:README.DOC"
```

opens the file "README.DOC" from the *default drive*, and

```
OPEN #1, 6, 0, "DCAR:*. *"
```


opens the X cartridge directory. The command

```
LOAD "DB:TEST.BAS"
```

loads the program "TEST.BAS" from drive 2, and

```
LIST "DPRN:"
```

lists your program to the printer (of course you could have used "P:" instead of "DPRN:").

The above examples showed how to use the SpartaDOS X "kernel" through the CIO "D:" device. Of course you still have all the other standard CIO devices at your disposal (e.g. "E:", "P:", "C:", "K:", "S:", etc.). The point is that through the command processor you may only access the SpartaDOS X "kernel" devices, but through the CIO you may access both sets of devices.

Default Drive and Directory The default drive and directory are the drive and directory the system uses when none are specified. Each drive has a default (or current) directory. To change the default drive, at the DOS prompt, simply type the new device identifier followed by a <RETURN>. For example

```
C:
```

sets the default drive to drive 3. To change the current directory on a drive, use the command CHDIR (or CWD). For example

```
CHDIR DOS
```

sets the current directory on the default drive to "DOS", and the command

```
CHDIR B: BASIC
```

sets the current directory of drive 2 to "BASIC" (assuming that the directories DOS and BASIC in these examples exist in the current directories).

Volume Names All SpartaDOS formatted diskettes have a volume name. They are used for two reasons

- To better organize your diskettes by naming them.

- For SpartaDOS to quickly tell the difference between the current diskette and the next diskette you put in the drive (since there is no way to tell when the door opens on the drive).

If you display a directory of an Atari DOS 2 format diskette, you will find that they are always named “AtariDOS” so that you can quickly tell the difference between diskette formats. Also, because there is no unique volume name on Atari DOS diskettes, the buffer system does not remember anything about the last access to these diskettes. Therefore, Atari DOS diskettes are much less efficient and are slower. In addition, loading binary files, especially those with many segments, can take *considerably* longer from an Atari DOS disk than from a SpartaDOS disk, sometimes many minutes more. It is recommended that files be copied to SpartaDOS disks or RAMDISKS before running.

Disk Format Compatibility SpartaDOS X has no problem reading from and writing to disks formatted and used with SpartaDOS 2.x and 3.x. It can also use a disk formatted with SpartaDOS 1.1, but, because of slight differences in format, it is recommended that these file be copied to disks formatted with SpartaDOS X to avoid problems when using them, later, with 1.1.

SpartaDOS 2.x and 3.x will also have no problem reading from or writing to disks formatted and/or written to by SpartaDOS X with one exception: any directory entries beyond the 126th will not be seen and may not be accessed. Deleting files before these in the directory will not allow them to be seen, since their physical position in the directory will not change.

SpartaDOS 2.x and 3.x will ignore the new file attributes (Archived and Hidden). They will not acknowledge these, nor will they change them.

Using External Cartridges with SpartaDOS X SpartaDOS X is in a “piggy back” cartridge, meaning that an external cartridge may be plugged into the top of the SpartaDOS X cartridge. If you are using an 800 cumputer, SpartaDOS X should be plugged into the left cartridge slot in the computer. If you have a 130XE connected to a Multi I/O, you may plug the SpartaDOS X cartridge into either of the slots on the connector.

If you have an R-Time 8 cartridge, you can plug it in almost anywhere, since it is not recognized by the computer as being a cartridge. You can plug it into the left (or only) slot and plug SpartaDOS X into it. You can plug it into the SpartaDOS X cartridge. You can also plug it into the right slot on an 800 or into the extra slot on the 130XE/MIO adapter.

If your computer has only one cartridge slot, you may plug the SpartaDOS X cartridge into the R-Time 8 or plug the R-Time 8 into the SpartaDOS X cartridge.

Any external program cartridge must be plugged into the top of the SpartaDOS X cartridge for the system to perform properly. Language cartridges (such as BASIC XL, BASIC XE, Action!, and MAC/65) may be left in the top of the SpartaDOS X cartridge until you wish to use some other cartridge. SpartaDOS X allows you to enable and disable these cartridges on demand. You can even turn off SpartaDOS X and the external cartridge to boot a game or another DOS without removing either cartridge.

Most game cartridges take control of the system during the boot process, preventing SpartaDOS X from initializing. This will not keep the game from operating properly, but it will keep you from using SpartaDOS X while the game cartridge is installed.

The Command Processor — Commands

The description of command processor is broken down into the two chapters. The first (“The Command Processor — Commands”) is a listing of SpartaDOS X commands in alphabetical order. The description of each includes the purpose, syntax, and type of command. The second (“The Command Processor — Advanced Features”) discusses batch files and I/O redirection, and contains more detailed information on some of its more complex features.

All the commands which follow are documented in alphabetical order. Command names and parameters represent their function or purpose so they should be easy to remember. For each command we briefly define the “**Purpose**” so you can quickly get an idea of what it is used for. We then show the “**Syntax**”, which shows the proper usage of the command along with its options if applicable. The following conventions are used in the “**Syntax**” section:

- [...] The parameters in the brackets is optional.
- a|b|...|z One or more of these options may be selected.
Refer to the specific command’s remarks for details.
- d:** Drive number or letter (A:...I;, 1:...9;, D1:...D9;, etc.).
- path** The path from the current or root directory to the desired one, such as TELECOM>EXPRESS> or \DOS\
- fname** The one to 8 character filename. With many commands, wildcards (* and ?) are allowed. Refer to the remarks for specifics for each command.
- .ext** A 0 to 3 character file extension. wildcards are often allowed.
- +, -, /** The characters should be used as shown.

SpartaDOS X Reference Manual

If there is an **“Alias”** for the command, it is shown next. We tend to have an alias when there is a shorter command which seems logical, to remain compatible with older SpartaDOS versions, or to try and maintain command similarities for people who use MSDOS.

The **“Type”** will either be **“internal”** or **“external”**. Internal commands are internal to the command processor itself — they require no other program to perform the command. External commands are found in the **“CAR:”** directory or may reference one of those files. 48K (6 banks) of the SpartaDOS X cartridge is devoted to these external commands.

“Related” commands are shown next. These may be in the same class or family of commands, or may include other ways of accomplishing the same function.

“Remarks” include all the details and special rules of command usage. The remarks may also show usage examples. A good way to learn SpartaDOS is to read through each command thoroughly and then try typing in examples of the command including its options. This will help you to understand SpartaDOS, which is important if you wish to be a SpartaDOS user.

If you have used a prior version of SpartaDOS, you will find the command processor similar in feel and will recognize most of the commands. Also, you will notice that the command processor has been greatly enhanced with more sophisticated batch files, command line I/O redirection, user definable prompts, command search paths, and more.

Now, on to the commands . . .

ARC (Archive Files) Command

Purpose Create and maintain file archives.

Syntax ARC command[option] [d:][path]arcfname[.ext] {filelist}

Type External — on device CAR:

Remarks SpartaDOS X brings a full featured ARC utility to Atari 8-bit computers. ARC is based on and compatible with ARC.EXE by System Enhancement Associates which was written for the IBM PC. It is also compatible with ARC versions running on the Atari ST and other computers. ARC will take a group of files and quickly combine and compress them into a single archive file, taking up far less disk space. It will also add or extract files to or from this archive, show a directory of the archived files, display the contents of an archived file, show the compression method used, encrypt/decrypt files, and more. “ARC” with no parameters will display the syntax, command list, and options.

The “**arcfname**” is the file name of the archive. The “**filelist**” is the list of files to be added ,deleted, updated, extracted, etc., to or from the archive. Leave a space between each filename in the filelist. Wildcards are perfectly legal. If no filelist is entered, “*.*” is assumed.

“**Command**” is one of the following:

- A Add file(s) to the archive. Add all files from the file list to the archive.
- M Move file(s) to the archive. Move deletes the source file once it has been added to the archive.
- U Update file(s) in the archive. Update will look at the date of the files in the archive, replacing files with a newer date, and add all files (from filelist) which do not currently exist in the archive.

- F Freshen file(s) in archive. This is the same as update but without the “add” feature. Freshen will replace the older files in the archive with any newer files of the same name.
- D Delete file(s) from the archive. Delete will remove the files listed in the filelist from the archive.
- X,E Extract file(s) from the archive. Both allow you to extract files from an archive. The method(s) of file compression used when creating the archive is reversed and the files specified in the filelist are restored to their original state.
- P Print file(s) to the screen. This allows you to examine the contents of files within an archive without extracting them. Of course this can be diverted to other devices with redirection; for example,

```
ARC P MYARC READ.ME >>PRN:
```

will divert the contents of “READ.ME” from the archive “MYARC” to your printer.
- L List file(s) in archive. This shows the filename, original file length, and date/time created of each file in the archive as well as the number of files and total size of files if extracted.
- V Verbose list of file(s) in archive. This command shows the filename, original file length, number of files, and total size, just as the L command does. Instead of date and time created, however, the V command shows stowage method, stowage factor (percent of space saved), the file size now, and the total size now.

Valid options are:

- B Retain a backup copy of the archive. This is a safety option for the A, M, U, F, and D commands. The B option will result in a backup of the old archive with the extension of “.BAK” as well as the new archive.

- S Suppress compression. This will archive files without compressing them. Most people will not use this option but it is faster than using compression.
- W Suppress warning messages. Use this command sparingly if at all. This will prevent those unsightly errors from being displayed but will also prevent mistakes from being discovered and avoided.
- N Suppress notes and comments. This will suppress the display of the standard ARC screen output which shows the current file being compressed or extracted, the compression method used, etc.
- H High speed. With the screen off on the Atari, processing speed is increased 20% to 30%. If you wish to go faster but don't need to see the screen, use this option. The screen display will return when finished.
- G Encrypt/decrypt an archive entry. This prevents others from reading your files. G must be the last option and must be followed by a password. If you forget your password, you will not have a useful archive. For example,

```
ARC AHGICD STUFF WASTE.DOC WASTE.COM READ.ME
```

In the preceding example the three files in the file list would be added into the archive called "STUFF.ARC" under the password of "ICD" with the screen off.

Archive entries are always saved in alphabetical order. This sorting function puts a practical limit of about 80 files per archive on 64K machines (USE OSRAM) and 180 files per archive on computers which use the extended memory mode (USE BANKED). ARC will not run on 48K machines unless they have an AXLON compatible memory upgrade installed. Archive entries do not save pathnames which means duplicate file names are not allowed (one will replace the other).

ARC is very useful for saving time while uploading/downloading files with a MODEM and saving space for archival storage. ARC uses four stowage methods and automatically determines the best method(s) suited to each file. Our SpartaDOS X version of ARC is also fully compatible with ALFcrunched files, but it is highly recommended that you unARC an ALFed file and then ARC it before adding or updating. This will assure the most compact compression and arrange all files alphabetically within the archive. The four stowage methods used in ARC are as follows:

Stored — no compression used. This is mainly used with very short files.

Packed — Strings of repeated values are collapsed. All files are packed before other compression methods are attempted.

Squeezed — Huffman encoding compression. This is usually only effective with larger machine language files. Huffman encoding uses a weighted binary tree method assigning the lowest bit representations to the most commonly used characters.

Crunched — Dynamic Lempel-Ziv compression with adaptive reset. This is created on the fly and is stored as a series of bit codes which represent character strings. Crunched is one of the more effective methods used. ALFcrunch exclusively uses a variant of this method.

NOTICE: The name ARC, compatibility, and all other similarities to the ARC.EXE program by SEA (for MSDOS computers) are intentional. This trademark and the “look and feel” of the program have been licensed for SpartaDOS X by ICD, Inc. from SEA (System Enhancement Associates).

ATR (Attributes) Command

Purpose Sets/clears file attributes in the directory. Replaces the Protect and Unprotect functions from older SpartaDOS versions.

Syntax ATR [+A|HIP] [-A|HIP] [d:][path]fname[.ext]

Alias ATTRIB

Type Internal

Related DIR

Remarks SpartaDOS X adds two new attributes to the standard SpartaDOS directory entry — these are the hidden and archived bits. The old commands PROTECT and UNPROTECT were used to set or clear the protection bit. With SpartaDOS X, the ATR command replaces the old commands and works with the new attributes.

Although many other commands allow the usage of the “S” (subdirectory) attribute, it is illegal to attempt to change this status bit as it would corrupt the subdirectory integrity. Therefore, ATR does not affect this.

Note that although the syntax of the ATR command looks similar to that of the DIRECTORY or TYPE commands, the attributes here are not the scan mode, but describe the set(+)/clear(-) attributes operation to be performed on the directory entry that matches the given filespec. This means that the scope of the ATR command is all files matching the filespec (including those files which are hidden).

The directory entry attributes are as follows:

- A Archived file. This attribute is cleared whenever a file is created or updated. The archive bit is set when the file is backed up by a program such as FlashBack!. This attribute is *not* related to the ARC command.
- H Hidden file. You may hide files and/or subdirectories. If a file is hidden, you may load it as a command only — commands such as TYPE and COPY will not see hidden files (unless you specify attributes with those commands). The file is hidden when this bit is set.
- P Protected file. You may not ERASE, or update protected files. Use the ATR command to protect or unprotect files. The file is protected when this bit is set.
- S Subdirectory. This attribute is unchangeable — thus not legal in the ATR command! This bit is set to indicate a subdirectory. If cleared, it would be seen as a file which could cause significant damage.

For example, to set the archived status and clear the protection bit of all “.COM” files, type the command

```
ATR +A -P *.COM
```

For further information about which status bits in the directory entry are affected by these new attributes, refer to the “Technical Information” chapter.

BASIC Command

Purpose This command enters the *internal* BASIC in your XL or XE computer (1200XL does not have internal BASIC).

Syntax BASIC [/N] [d:][path][fname] [parameters]

Type External — uses CAR.COM on device CAR:

Related CAR, SET

Remarks If no filename is given, control is given to the *internal* BASIC of your XL/XE computer. If you do specify a filename, the internal BASIC is enabled and the **binary** file you specified is loaded and run. The optional “parameters” are whatever the program “fname” needs. The “/N” option returns to BASIC after running “fname”, instead of the command processor which is the default. To automatically load and run a BASIC program from the command processor, read the I/O Redirection Section in the “Advanced Features” chapter.

This command is recognized by the command processor as an internal command that chains to the external program “CAR.COM”, so both the CAR and BASIC commands share the same external program. CAR.COM is memory resident while you are in the BASIC environment, so MEMLO will be slightly higher during this period.

SpartaDOS X has a MEM.SAV facility somewhat like that of Atari DOS 2, but much more powerful. The environment variable “BASIC” should be set to the file you wish to use as the memory-save file for BASIC. If no such environment variable exists, then the memory-save feature is disabled. If this feature is disabled, BASIC will be entered cold (there will be no program in user memory).

The default value of the “BASIC” variable is “I:>BAS.SAV”. You of course may change this with the SET command, for example:


```
SET BASIC=D8: BASIC.SAV
```

sets the variable to “D8: BASIC.SAV”. To see the current value of “BASIC” (and all the other environment variables) just type:

```
SET
```

and to clear the variable (i.e. disable the BASIC memory-save feature) type

```
SET BASIC
```

(See the SET command for a further explanation.)

With the memory-save feature enabled, if a problem is encountered when loading or saving the memory file (BASIC.SAV by default), an error message will appear. If this happens while loading the memory file, you will be prompted with the old MEMLO (when the file was saved). If an error exists while saving the memory file, you will be notified of that. In either case, you will have the option to abort and correct the problem or to proceed, deleting the memory file. Press the <ESC> key if you wish to abort or <RETURN> to proceed. More detail of the two situations that can occur follows:

- Upon entering BASIC, the current MEMLO does not match the MEMLO in the memory-save file. This can occur after installing extra drivers since last time you entered BASIC (such as the keyboard buffer, RAMDISK, etc), or LOADING commands such as X or COMMAND (see the LOAD command). At this point you may press <ESC> and restore the system to the way it was when you last entered BASIC (By COLD starting and/or LOADING programs), or press <RETURN> and enter BASIC cold. This will also happen if the memory-save file has somehow been corrupted.
- Upon exiting BASIC (using the DOS command), the disk fills up or is not online and the memory-save file can't be saved. You have the option to go to DOS (RETURN) and lose the current BASIC program in memory, or to go back to BASIC (ESC) and SAVE whatever you were working on or clear up the disk problem.

In addition to saving the contents of user memory, the memory-save feature saves page 0 (from \$80-\$FF), and pages 4-6. This means that you may alternate between BASIC and CARtridge without losing what you were working on. When you enter BASIC the memory-save file is loaded, allowing you to edit a BASIC program, go to DOS, reboot the computer, and enter BASIC with exactly what you were working on before rebooting the system (as long the memory-save file is present and valid).

Performing a cold start (a jump to \$E477) while in BASIC will cause the SpartaDOS X cartridge and the external cartridge plugged into the SpartaDOS X cartridge, if any, to be disabled. This will have the same effect as typing COLD /N from the command processor.

BOOT Command

Purpose	This command tells a SpartaDOS formatted disk to boot a particular program at start up (normally a disk-based DOS).
Syntax	BOOT [d:][path]fname[.ext]
Type	Internal
Related	COLD, FORMAT
Remarks	The DOS loader on the first three sectors of each SpartaDOS formatted diskette (version 2 and above), can load and run files in the same manner as a command file. Normally DOS is loaded, but anything could be loaded as long as it avoids the loader memory (\$2E00-\$3180).

The FORMAT command does not put SpartaDOS on the diskette it formats, so, if you want to create a bootable SpartaDOS diskette, you must COPY SpartaDOS to the diskette and use the BOOT command. If you just use SpartaDOS X, this will never be necessary (since SpartaDOS X boots from cartridge), but if you have SpartaDOS 3.2 (or 2.3), you may wish to create bootable diskettes. Of course you may still use the XINIT command to format and install DOS on your diskette.

This command is the most misunderstood command in SpartaDOS, so here are a few pertinent facts you should know:

- The BOOT command simply writes the starting sector number of the sector map of the file to boot in a specific location on sector 1 of the diskette. (See “Technical Information”)
- If the file which is set to boot is either ERASEd or COPYed over, the boot flag is cleared — you will get the message “Error: No DOS” when attempting to boot that diskette until you set a new file (e.g. X32D.DOS) to boot!
- The file you set to boot may reside anywhere on the diskette — even in a subdirectory.

- This command does **not** work with SpartaDOS 1.1 — 1.1 does not contain a 3 sector booter! SpartaDOS 1.1 has a simplistic booting scheme which just loads a number of consecutive sectors. Since you now have SpartaDOS X, we *strongly recommend* that you abandon SpartaDOS 1.1.

CAR Command

Purpose This command enters the cartridge plugged into the top of the SpartaDOS X cartridge. If a filename is specified, then that **binary** file is loaded and run with the cartridge enabled.

Syntax CAR [/N] [d:][path][fname] [parameters]

Type External — on device CAR:

Related BASIC, COLD, SET

Remarks If no filename is given, then control is given to the cartridge plugged into the SpartaDOS X cartridge. If you do specify a filename, then that **binary** file is loaded and run with the cartridge enabled. This is useful for compiled Action! programs that need to call routines within the cartridge. The optional “parameters” are whatever the program “fname” needs. The “/N” option returns to the cartridge after running fname, instead of to the command processor which is the default.

This command is recognized by the command processor as an internal command that chains to the external program “CAR.COM”, so both the CAR and BASIC commands share the same external program. CAR.COM remains memory resident while in the cartridge environment, so MEMLO will be slightly higher during this time. It will return to the lower value when the cartridge is exited.

SpartaDOS X has a MEM.SAV facility somewhat like Atari DOS 2, but with much more power. The environment variable “CAR” should be set to the file you wish to use as the memory-save file for the cartridge. If no such environment variable exists, then the memory-save feature is disabled. If this feature is disabled, then the cartridge will be entered cold (there will be no program in user memory).

The default value of the “CAR” variable is “I:>CAR.SAV”. You may change this with the SET command (see the BASIC and SET commands for details on this).

If a problem is encountered when loading or saving the memory file, you will be told the problem and asked if you wish to cancel (i.e. go back to where you came). Press the <ESC> key if you wish to cancel. The two situations that can occur are as follows

- Upon entering the cartridge, the current MEMLO does not match the MEMLO in the memory-save file. This can occur after installing extra drivers since last time you entered the cartridge (such as the keyboard buffer, RAMDISK, etc), or LOADING commands such as X or COMMAND (see the LOAD command). At this point you may press <ESC> and restore the system to the way it was when you last entered BASIC (By COLD starting and/or LOADING programs), or press <RETURN> and enter the cartridge cold. This will also happen if the memory-save file has somehow been corrupted.
- Upon exiting the cartridge (using the DOS command), the disk fills up or is not online and the memory-save file can't be saved. You have the option to go to DOS (RETURN) and lose the current data in memory, or to go back to the cartridge (ESC) and SAVE whatever you were working on or clear up the disk problem..

In addition to saving the contents of user memory, the memory-save feature saves page 0 (from \$80-\$FF), and page 4-6. This means that you may alternate between BASIC and CARtridge without ever losing what you were working on. Whenever you enter the cartridge the memory-save file is loaded, thus you can edit a program in the cartridge, go to DOS, reboot the computer, and enter the cartridge with exactly what you were working on before rebooting the system (as long the memory-save file is present and valid).

Executing a cold start while in the cartridge will cause SpartaDOS X to be disabled, while leaving the external cartridge enabled. This is the same as typing COLD /C from the command processor.

CHDIR (Change Directory) Command

Purpose This command changes the current (working) directory on the specified drive, or displays the current directory path if no path is given.

Syntax CHDIR [d:][path]

Alias CD & CWD

Type Internal

Related MKDIR, RMDIR, PATH

Remarks Directories (also called subdirectories or folders) are used like file folders to organize your files. They also make searching large storage areas (such as hard drives) much faster. In a file cabinet it is much quicker to go to a file folder and search through a few documents than a pile of all your documents. computers work the same way. It is much quicker for DOS to go to a subdirectory and search through a few files than it is to search through one long file list. CHDIR allows you to move among your directories.

The current directory is where SpartaDOS looks to find files whose names were entered without specifying a directory. If you do not specify a drive, the default drive is assumed. If you enter the CHDIR command with no parameters, the current directory path of the current drive is displayed. (This mode is the same as the ?DIR command from SpartaDOS 3.2.)

Whenever SpartaDOS is re-initialized (i.e. RESET), the default directory on *every* drive is reset to the MAIN (root) directory. The default directory of a drive is also reset to the MAIN directory if the diskette has been changed.

This command has no effect on MYDOS diskettes even though subdirectories are supported. This is due to the fact that there is no foolproof way to detect a disk change on DOS 2 style diskettes (SpartaDOS diskettes have a volume names, a random number, and a write count for disk change detection — see “Technical Information”).

CHKDSK Command

- Purpose** Show volume, free/total disk space, and sector size of the selected drive (or diskette).
- Syntax** CHKDSK [d:]
- Type** Internal
- Related** FORMAT, MEM, VER
- Remarks** The CHKDSK command is used to quickly see how much space is available on a drive and the sector size (this information is not available by doing a DIRectory). Note that the volume name of all Atari DOS 2 style diskettes will appear as "AtariDOS".

You will notice that the disk write-lock status is omitted — this feature is no longer supported. We found this to be more of a hassle than it was worth, and it did not protect you from formatting the diskette. (The write-lock feature of the Multi I/O still works and is totally independent — it is a far more secure write-lock.)

The following is a sample output of the CHKDSK command.

```
Volume: SPARTA_1 0A 25
Bytes/sector: 256
Total bytes: 184320
Bytes free: 123390
```

The two numbers following the volume name are used for disk change detection in cases where volume names are the same on both diskettes. The first is a random number generated when the disk was formatted. The second is a sequence number which is incremented each time a file on the disk is opened for write.

CHTD (Change Time/Date Stamp) Command

- Purpose** This command changes the time/date stamp on all files matching the given filespec to the current time and date.
- Syntax** CHTD [+A|H|P|S] [-A|H|P|S] [d:][path]fname[.ext]
- Type** External — on device CAR:
- Related** DATE, TD, TIME
- Remarks** By default, this command will only change the time/date stamp on non-hidden and non-protected files — this may be overridden. (See ATR command for more information on attributes.) You must enter a filespec since “*.*” is not assumed.

CHVOL (Change Volume Name) Command

- Purpose** This command changes the volume name on the specified drive.
- Syntax** CHVOL [d:]volname
- Type** External — on device CAR:
- Related** CHKDSK, FORMAT, DIR
- Remarks** This command will not change the volume name on Atari DOS 2 diskettes since they physically have no volume name. Up to eight characters are allowed on SpartaDOS formatted diskettes. The volume name may contain any ATASCII characters including spaces and inverse characters.

COLD Command

Purpose This command reboots the system (by doing a jump through \$E477).

Syntax COLD [/CN]

Type Internal

Related BOOT, CAR

Remarks This command is an alternative to switching the computer's power off and back on. The major advantage of using COLD is that the extended banks of RAM will retain their memory, thus the data in your RAMDISKS will still be there. (See the RAMDISK.SYS driver description.) This is equivalent to the SpartaDOS 3.2 command:

```
RUN E477
```

This command has two options, they are:

- C Reboot the computer with SpartaDOS X disabled and the cartridge plugged into SpartaDOS X enabled.
- N Reboot the computer as if there were no cartridges in your computer.

Hold down <OPTION> while pressing <RETURN> to reboot without internal BASIC.

Once SpartaDOS X has been disabled, it will be necessary to turn the computer off and back on to re-enable SpartaDOS X.

COMMAND (The Command Processor)

- Purpose** This program allows you to enter commands and run other programs. It is not entered as a command itself but is automatically invoked when you enter DOS.
- Type** External — on device CAR:
- Related** All Commands
- Remarks** Many of the commands are of type “internal” — this means that the command processor knows how to perform the command without loading any other programs.

“External” commands must load from disk or CARtridge into memory and then perform their function. When you execute these commands, they must reside on the current drive and directory, otherwise you must specify what drive or device they reside on (by preceding the command with a device or drive identifier). The PATH command can add additional drives/paths to search for the file. For example the default PATH (which is)

```
PATH CAR:
```

allows commands such as CHTD or DUMP to run without having to specify the “CAR:”. Of course you may add additional directory paths (see the PATH command).

You will notice that the command processor itself is “external”. This is to give you more memory (3-4Kbytes) to run your application programs. In fact, whenever you run an “external” command or program, the command processor is unLOADed from memory and replaced by the new program. Then, when that program is finished running, the command processor is reLOADed and awaits your command. The exception to this rule is if you enter the command

```
LOAD COMMAND.COM
```


This actually holds and links the command processor in memory, thus the unLOAD/reLOAD cycle is circumvented.

COPY Command

- Purpose** Copies one or more files to another drive and, optionally, gives the copy a different name if specified
- COPY also copies files to the same disk. In this case, you must give the copies different names unless different directories are specified; otherwise, the copy is not permitted. Concatenation (combining of files) can be performed during the copy process with the “/A” parameter.
- You can also use the COPY command to transfer data between any of the system devices. Some applications of this would be to create a batch file or to print a text file.
- To copy files with a single disk drive and no RAMDISK, see the MENU command. MENU has a provision for disk swapping, while COPY does not.
- Syntax** COPY [d:][path][fname][.ext] [d:][path][fname][.ext]/[A]
- Type** Internal
- Related** MENU, TYPE
- Remarks** The first filespec specified is the source file name. If none is given, a default filespec of “*.*” is assumed (which will copy all files in that directory). The device for the source file should be given; however, it is possible to omit the source device if you use commas (instead of spaces) to separate parameters, for example:
- ```
COPY, ,D3:
```
- will copy all files from the default drive/directory to the current directory of drive 3. The second filespec is the destination — if no filename is specified, a default filespec of “\*.\*” is assumed (which will copy the files without changing the names).

You may use wildcards ('\*' and '?') in both source and destination filespecs. If used in the pathnames, the first directory match will be used.

When using wildcards with the COPY command, the same renaming convention as in the RENAME command is used. The source filespec is used to find directory matches, and the destination filespec renames them by overriding characters in the source name with the non-wildcard character in the corresponding position of the destination name.

If you are copying from a device other than "DSK:" (alias "Dn:" or just "n:"), then just one file is copied and the destination filespec is the name that the file will be saved under. For example:

```
COPY CON: B:*
```

is illegal because you may not have wildcard characters in a destination filename when COPYing from a character device (or for that matter SAVEing any file). However, if copying from one character device to another character device, filenames are not used. (Character devices never use filenames.) For example:

```
COPY CON: PRN:
```

As in the above two examples, when COPYing from "CON:" you signal the end of file by pressing a <CTRL-3> after typing the text. Also, a <RETURN> must follow each line you enter, otherwise that line will be lost.

Another use for the COPY command is to list files to the printer or screen, for example:

```
COPY README.DOC CON:
```

will display the contents of "README.DOC" to the screen and:

```
COPY README.DOC PRN:
```

will send it to the printer. Note that both of the above examples could have been performed with the TYPE command as follows:

```
TYPE README.DOC
```

```
TYPE README.DOC >>PRN:
```

with the second command sending the contents of the file to the printer.

You may also append files using the COPY command by using a “/A” immediately following (no space) the destination filespec. (SpartaDOS 3.2 allows a “/A” when SAVEing any file to force append mode — SpartaDOS X only supports this feature on the COPY command.)

If you only have one drive and wish to COPY files from one diskette to another, you must either COPY the file from the source diskette to a RAMDISK and then from the RAMDISK to the destination diskette, or use the MENU program as it allows disk swapping during copying.



### DATE Command

---

- Purpose** This command displays the current date and allows you to set the date.
- Syntax** DATE
- Type** Internal
- Related** CHTD, TD, TIME
- Remarks** This command produces the following output
- ```
Current date is 6-24-88
Enter new date:
```
- You may enter the new date or press <RETURN> if you don't want to set a new date. Enter the date in the format "mm-dd-yy" where "mm" is the month, "dd" is the day, and "yy" is the year.
- This command will produce meaningless results if you do not have a clock driver installed in your system. The two clock drivers are "CLOCK.SYS" and "JIFFY.SYS" — the first using the R-Time 8 and the second using the jiffy counter to keep its time. By default, one of these drivers will be installed when you boot, but this can be overridden by creating a custom "CONFIG.SYS" file and not including these drivers in the configuration.

DIR (Directory) Command & DIRS (Short Directory) Command

Purpose Lists either all the directory entries, or only those matching a specified filespec. DIR will optionally give you a count of files listed.

Syntax DIR [+A|H|P|S] [-A|H|P|S] [d:][path][fname][.ext] [/PC]
DIRS [+A|H|P|S] [-A|H|P|S] [d:][path][fname][.ext] [/PC]

Type Internal

Related ATR, FIND, MENU, PATH, PAUSE, PROMPT

Remarks DIR displays the SpartaDOS file directory showing filename, extension, file size in bytes, date, and time created. It also shows a <DIR> in the size field when it sees a subdirectory, displays the Volume and Directory name at the top of the listing, and shows the Free Sectors count at the end of the listing. If you include a “P” parameter, the DIR command will wait for a key press after displaying each directory screen (23 lines). The “C” parameter will give a count of the number of entries displayed in that directory.

When reading an Atari DOS 2 type diskette, the date and time are omitted for obvious reasons and the file size is roughly converted to bytes. (Atari DOS 2 and clones use sector lengths instead of bytes in the directories so this can not be an exact file size representation.) All AtariDOS 2 type diskettes will have a volume name of “AtariDOS” and a directory name of “ROOT”.

You may specify the attributes of the files you wish to display, for example

```
DIR +S
```

will display only subdirectories. Note that the default directory attributes (no attributes specified) is “-H” (do not show hidden files). If you wish to see all files (including hidden files), simply enter

```
DIR +
```

Using a '+' with no attribute listed will match all files, regardless of attribute. This will work with any command that allows attribute selection.

The DIRS command has exactly the same syntax, but it displays the directory in Atari DOS 2 “compatibility mode” — with no time/date, and with the file size displayed in sectors rather than in bytes. Since the Free Sectors count in DIRS is limited to three digits, the maximum size displayed will always be 999. It also displays the “protected” status (+P) as “*” before each protected filename.

The attributes are as follows:

- A Archived file. This attribute is cleared (-) whenever a file is created or updated. It is set when the file is backed by a program such as FlashBack!
- H Hidden file. You may hide files and/or subdirectories. If a file is hidden, you may load it as a command only. Commands such as TYPE and COPY will not see hidden files (unless you specify attributes with those commands).
- P Protected file. You may not ERASE or update protected files. Use the ATR command to protect or unprotect files.
- S Subdirectory. This attribute is unchangeable.

If you do not specify a filespec, “*.*” is assumed as in the following examples:

```
DIR MYSUB>
DIR +P
DIR ..\
```

Note that you must follow a subdirectory with a “>” or “\” character if you wish to see the contents of that directory.

DUMP (Display File in HEX Format) Command

Purpose This command displays a file in HEX and ATASCII form.

Syntax DUMP [d:][path]fname[.ext] [start] [len]

Type External — on device CAR:

Related TYPE

Remarks The parameters “start” and “len” are the start addresses in the file and the number of bytes to dump (respectively). They are assumed to be in decimal format unless preceded by a “\$” (in which case they are HEX format).

DUMP is useful to quickly examine the contents of a file. To modify a file or examine and modify disk sectors, use DiskRx from SpartaDOS Tool Kit.

ERASE Command

- Purpose** This command deletes the file in the specified directory on the designated drive, or deletes the file from the default drive if no drive is specified. If no path is specified, the file is deleted from the current directory.
- Syntax** ERASE [d:][path]fname[.ext]
- Alias** DEL & DELETE
- Type** Internal
- Related** MENU, UNERASE
- Remarks** You can use wildcards such as '*' and '?' to delete multiple files, but use caution since no warning is usually given. If you do enter *.* as the specifier, SpartaDOS X will prompt you with:
Erase ALL: Are you sure?
- Any other combination of wildcards and characters will assume you know what you want. Use the VDEL command (in SpartaDOS ToolKit) for a prompted delete or use MENU for tagging files to delete.

FIND (Find Files) Command

- Purpose** This command searches all directories on all drives for files matching the given filespec. If you enter a drive number, FIND will only look on that particular drive.
- Syntax** FIND [d:]fname[.ext]
- Type** External — on device CAR:
- Related** DIR
- Remarks** FIND will quickly find a file anywhere on your drives. This becomes very useful when you start using subdirectories and multiple drives. FIND works like WHEREIS.COM (from SpartaDOS Tool Kit) with a few exceptions. FIND's search is slightly different and it does not have a parameter to display filesize, time, and date, for each file found.
- The filename may include wildcards as desired. All filename matches found will be displayed with the full path from the root directory to the filename match. The number of matches found will be displayed at the end of the search. All drives will be searched unless a drive number is specified. FIND will also find and display hidden filenames.

FORMAT Command

- Purpose** Initializes a disk in either SpartaDOS or Atari DOS 2 format. You may select density, sector skew, tracks, and volume name before formatting. It supports most known hardware configurations for your computer.
- Syntax** FORMAT
- Type** Internal
- Related** BOOT, CHVOL
- Remarks** The FORMAT command is really a menu driven program which allows you to initialize just about any type of disk that works with an Atari 8-bit computer. It can be called from the command processor by typing FORMAT or from within a program with XIO 254 (see the “Programming with SpartaDOS X” chapter). This allows FORMAT to be used with most programs that support disk initialization like AtariWriter or 850 Express!
- When you format a floppy diskette, you are first writing the sector structure to the diskette so the DOS has a place to put the program information. Next the directory structure is written to the diskette which is where the DOS keeps track of sector usage. You can also initialize a RAMDISK or Hard Disk in the FORMAT menu but only with the BUILD DIRECTORY option.
- You can exit or quit the FORMAT menu at any time before formatting begins by pressing <ESC>.
- After entering the FORMAT menu you choose the following parameters:

- U Unit** is the initial selection that must be made. The formatter needs to know which drive you wish to initialize. Valid choices are: 1 - 9 or A - I. After entering the unit number or letter the program reads the drive to determine what type it is. FORMAT automatically determines whether the drive is a floppy disk drive and if it is configurable or if the drive is a RAMDISK/Hard Drive which appear the same at this point.
- Note:** FORMAT will only write directories to RAMDISKS and hard drives. An internal RAMDISK must be installed with the RAMDISK.SYS driver. A hard drive partition must be low-level (physical) formatted with with a program that should have been provided with the hardware.
- S Skew** refers to the order in which the sectors are arranged in a given track. The two valid choices are: High Speed and Standard. High Speed will automatically put the correct UltraSpeed skew on a disk using SpartaDOS with the US Doubler or Indus GT drives. It will also put the correct high speed skew on the Atari XF551 drive under Double Density. Standard skew is used on all other floppy drives. If you do select High Speed skew and the drive does not support a high speed mode, the format program will receive an error from the drive and then try to format under Standard skew. The correct skew is required on most drives for the fastest possible reading and writing. Skew is not applicable to RAMDISKS and Hard Drives, since they can not be physically formatted by this program.

NOTE: Skew refers to the order in which the sectors are written on the diskette. The optimum skew will position the sectors so that after sector 1 is read and the drive CPU is ready for the next, sector 2 is directly under the head for reading; after 2 is read, 3 is directly under the head, etc. (Usually 2-8 sectors have passed under the head before the next sector can be read. This varies with drive speed and SIO baud rate.) If the skew is off, it may take a full disk revolution to read or write the next sector each time. No harm is done, the drive just reads and writes slowly.

- M Mode** is either **Sparta** or **Atari**. Sparta is for SpartaDOS disk directory structure and Atari is for all the AtariDOS 2.0 clones and their directory structures.

NOTE: FORMAT does not write a "DOS" file to the diskette. If you want to create a bootable SpartaDOS disk, you must copy a "DOS" file from the SpartaDOS Construction Set to your formatted disk and then use the BOOT command. (See the BOOT command.) If you want to create a bootable AtariDOS diskette, you will need to boot AtariDOS and write the DOS.SYS and DUP.SYS to the drive through its menu. SpartaDOS X will boot with any or no disk.

- V Volume** is a way of naming your diskettes for organizational purposes. Up to eight characters are allowed on SpartaDOS formatted diskettes. The volume name may contain any ATASCII characters except spaces. Volume is used on SpartaDOS diskettes only and is not applicable to other DOS types.

- D Density** may be one of three types used with 8-bit Atari computers. These are: **Single**, 128 bytes per sector FM, **Dual**, 128 bytes per sector MFM, or **Double**, 256 bytes per sector MFM. (FM and MFM refer to bit density where MFM writes twice the number of bits in the same area as FM.) Atari 810s only support Single density. Stock Atari 1050s support Single and Dual. Atari 1050s with the US Doubler (or other OS enhancement), Atari XF551s, and Indus GTs support all three densities. Most other disk drives for the 8-bit Atari support Single and Double density.
- T Tracks** can be **40 SS**, **40 DS**, **77 SS**, **77 DS**, **80 SS**, and **80 DS**. SS means Single Sided (1 head writes on one side of the diskette) and DS means Double Sided (2 heads with each writing on opposite sides of the diskette). All Atari brand drives will use 40 SS except for the XF551 which is capable of 40 DS. Most of the other 5 1/4 inch drives will be either 40 SS or 40 DS. (See your drive manual if you are not sure.) 77 Tracks is used for 8 inch disk drives connected with an interface like the ATR8000 or PERCOM controller. 80 Tracks is used for 3 1/2 inch drives and high capacity 5 1/4 inch drives with a similar interface. All drives with two heads will also format in the SS mode.

NOTE: The drive controllers do not provide adequate feedback to the computer when formatting a diskette to determine whether the Tracks selection is wrong for the drive. It is important to enter the correct information or the disk will end up with an incorrect free sectors count.

- F **Format Disk** will start the physical format of a floppy diskette assuming you have entered all the other parameters required. It also writes the directory structure selected in Mode after the physical format and verify is completed. (The physical format and verify are functions of the floppy disk controller and not affected by the SpartaDOS VERIFY command.) **CAUTION:** The Format Disk procedure obviously destroys all previous information stored on the diskette.
- B **Build Directory** is the initialization option available for RAMDISKS and Hard Disks, although it will work equally well with floppy disks. The only parameters available for these disks are Unit number and Volume name. The others are predetermined or not applicable. Build Directory writes fresh SpartaDOS directory structure to the drive Unit selected which means it will destroy all previous information stored in the RAMDISK or Hard Disk partition.

The physical format of a Hard Disk drive must be performed by a special program written for the particular hard drive, interface, and controller. That is considered a low-level format and is beyond the scope of the FORMAT menu. The physical format of the RAMDISK is provided by the RAMDISK handler at installation.

Sectors and Bytes counts are also shown on the FORMAT menu and are determined by what is read from the configuration on RAMDISKS or Hard Disks or by the parameters selected for a Floppy Drive format.

Indus GT NOTES: It should be mentioned here that the Indus GT has a few known quirks or bugs. Because of this, Dual (Enhanced) Density is not supported on this drive. This should not be a problem since Dual Density is really unnecessary on a double density drive. The Indus has also been known to keep spinning indefinitely when used in a system with US Doubler enhanced 1050s. If you experience this problem, the best solution is to stop using mixed drives in your system.

KEY (Keyboard Buffer) Command

- Purpose** This command installs a 32 character keyboard buffer and also links an “internal” KEY command into your system (for turning the buffer on and off).
- Syntax** KEY ON|OFF
- Type** External — on device CAR:
- Remarks** The first time you use this command, it installs a keyboard driver into your system. The keyboard buffer will provide a faster key repeat and allow you to type ahead while the system is busy. Then the ON/OFF parameter is interpreted, enabling or disabling the keyboard buffer accordingly.
- Once the keyboard buffer has been installed, the global symbol “@KEY” is defined and further KEY commands call this symbol to turn the buffer on and off.
- NOTE: The keyboard buffer may be incompatible with some programs but is more compatible with other programs than the SpartaDOS 3.2 buffer (most notably with the Action! cartridge).

LOAD Command

Purpose	Loads a file (does not run). This is useful for keeping commonly used commands resident in memory, thereby eliminating the need for these commands to load from disk. If no filename is used, all files previously loaded are removed from memory.
Syntax	LOAD [d:][path][fname][.ext]
Type	Internal
Related	MEM, SAVE
Remarks	<p>If you LOAD a standard binary load file, the results are identical to those achieved with SpartaDOS 3.2. The file is loaded into memory and not run. There are a few primary uses:</p> <ul style="list-style-type: none">• To load MAC/65 object files into memory and then SAVE them back as continuous non-segmented binary files.• To load a binary program prior to running a debugger (for testing purposes).

The only difference is that, if the program contains an INITAD segment, that *will* be executed.

One use of LOAD is to temporarily make external commands memory resident. This will only work with special SpartaDOS X relocatable external commands. See the MEM command for more details.

LOAD is used to:

- Keep an external command such as CAR or X, or keep the command processor (COMMAND.COM) resident in memory.
- Remove all non-installed commands or programs from memory (use LOAD with no filename).
- Load a subprogram into memory for use by other commands.

MEM Command

Purpose This command displays the current low memory limits of your system and the number of available banks of windowed RAM.

Syntax MEM

Type Internal

Related CHKDSK, LOAD

Remarks This command displays the bottom limit of available user and extended memory. Two limits for each memory region are given — the first being the top limit of installed drivers and the second being the top limit of held-in-memory applications.

In the following example,

```
Main: $0F6D,$1456
     Ext: $6123,$6455
     7 banks of RAM available
```

the installed drivers use memory from \$700-\$F6C and \$4000-\$6122, and the held-in-memory applications used reside in memory from \$F6D-\$1455 and \$6123-\$6454.

The held-in-memory applications are LOADED into memory and consist of files such as COMMAND.COM and X.COM. The drivers installed in memory are files such as SPARTA.SYS, ATARIDOS.SYS, RAMDISK.SYS, etc.

Normally the first and second numbers above will be the same. The OS variable MEMLO (at \$2E7) contains the second number. If LOAD is executed with no parameters then all in-memory applications are abandoned and the second number is lowered to the first.

NOTE: If a permanently installed driver is installed after LOADING a held-in-memory application, both low memory values will be raised above it and any held-in-memory applications will become permanent.

Although there are two possible extended memory regions, only one may be used by SpartaDOS X. This is determined at bootup time and is dependent upon the CONFIG.SYS file and/or the computer you are using. (See “Technical Information” for a discussion on extended memory.) Note that although the MEM command does not explicitly say what memory extended memory range is in use, it can be inferred by the addresses listed in the “Ext:” field. The ranges are as follows

- \$4000-\$7FFF Banked RAM (130XE or extended RAM computer)
- \$E400-\$FFBF OS RAM (not available on the Atari 800 computer)

The “banks of RAM available” field indicates how many banks of RAM are still available for a RAMDISK driver and/or BASIC XE extended mode. **Note that you must have at least 4 banks available for BASIC XE extended mode.** Failure to pay attention to this fact may cause your system to crash (generally at the very worst time).

MENU Program

Purpose This program allows you to select files and then perform COPY, ERASE, RENAME, etc. commands on all selected files. It is similar to other SpartaDOS menu programs, but provides many new features.

Syntax MENU

Type External — on device CAR:

Related COPY, ERASE, RENAME, TYPE

Remarks MENU is useful for operations that include more than one file and is required for single drive copies. It includes three main windows with the commands and prompts displayed below the windows. The “upper left window” is for directories. It will display the subdirectories along with the tree structure showing how they are related. The “upper right window” shows statistics on the area logged. This includes: filespecs, total files, total bytes, tagged files, and tagged bytes. The “lower window” shows the files.

The command menus are broken up into three major classifications: file (**File**), directory (**Dir**), and extra (**Xtra**). The classification is indicated at the lower left of the screen. Toggle between the file and directory command menus by pressing <RETURN>. You can go to the extra command menu by pressing <ESC>. To exit from the MENU you press <ESC> then “Q”. The “^” character before a menu selection key means to hold down the CONTROL key while pressing the selection key.

File command menus include: COPY, DELETE, FILESPEC, LOG, PRINT, RENAME, TAG, UNTAG, and VIEW. **Dir** command menus include: AVAILABLE, DELETE DIRECTORY, FILESPEC, LOG, MAKE DIRECTORY, PRINT, TAG DIRECTORY, and UNTAG DIRECTORY. **Xtra** command menus include: DISPLAY, QUIT, and SORT.

File Commands

While in the file command menu, use the “↑↓” keys to move the file selector up or down one file at a time or use the “← →” keys to move the file selector up or down one screen at a time. The files shown in the file window are sorted alphabetically by name. They represent the current directory shown in the directory window under the directory selector. (See “Dir Cmnds”.)

- C **Copy** — will copy the file under the file selector. You will be prompted for a destination drive, then a destination path. If copying to the same drive you will also be prompted to insert the destination disk, and then to insert the source disk.
- ^C **^Copy** — will copy all tagged files. Prompts are the same as with Copy.
- F **Filespec** — allows you to enter a filespec with wildcards to narrow down the logged (and displayed) files. Only legal filename characters and wildcards are allowed. Do not enter drive number or path here; instead use Log for that.
- L **Log** — will allow you to change the drive number logged and/or path.
- P **Print** — will print the file currently under the file selector. This is only useful for ASCII text files unless you have a printer driver installed which will print ATASCII graphics characters.
- ^P **^Print** — will print the files currently tagged. It will send a form feed in between files.
- R **Rename** — will allow you to rename the file under the file selector. As a reference, rename prompts you with the present drive number, path, and filename. You can then enter the new filename directly under the old.
- T **Tag** — will tag (mark) the file under the file selector then move the file selector down one filename. A small tag character “◆” will appear to the right of the filename showing it as tagged.

- ^T ^Tag** — will tag all files currently logged (in the current directory).
- U Untag** — will untag the file under the file selector. The tag character will disappear and the file selector will move down to the next file.
- ^U ^Untag** — will untag all files currently logged (in the current directory).
- V View** — will display the contents of the file under the file selector.

Dir Commands The directory selector indicates the current directory. While in the directory command menu, use the “↑↓” keys to move the directory selector up or down one directory at a time. When finished with the “Dir Cmnds”, press <RETURN> to go back to “File Cmnds” or <ESC> for “Xtra Cmnds”.

- A Avail** — will give you the amount of free space available on a drive. You are prompted to enter a drive number and will be shown the free space in bytes.
- D Del Dir** — use “↑↓” to move the directory selector. If the directory selected is empty (as shown in the file window), it can be deleted.
- F Filespec** — allows you to enter a filespec with wildcards to narrow down the logged (and displayed) files. Only legal filename characters and wildcards are allowed. Do not enter drive number or path here; instead use Log for that. You must go to the “File Cmnds” if you want to do any file operations other than Tagging or Untagging full directories.
- L Log** — will allow you to change the drive number logged and/or path.

- M **Make Dir** — will create a new subdirectory in the current directory selected. After the new directory is created, the system will relog and you will be back at the root of what was previously logged (always indicated by “>”).
- P **Print** — will prompt you with two choices: Directory or Tree. Directory will print the list of the files as displayed in the file window. (If the display is set to show the short form files, they will be printed in one long list, not side by side as displayed.) Tree will print the directory map (tree structure) as displayed in the directory window.
- T **Tag Dir** — will tag all files in the current directory (under the directory selector).
- ^T **^Tag Dir** — will tag all files in all directories currently logged.
- U **Untag Dir** — will untag all files in the current directory (under the directory selector).
- ^U **^Untag Dir** — will untag all files in all directories currently logged.

Xtra Commands

The “Xtra Cmnds” always take you back to the previous command menu when finished (except Quit). You can also press <ESC> to leave this menu.

- D **Display** — toggles the display in the file window between two types. The default display shows the filename with extension, the status of the three file attributes, the file size in bytes, along with the date and time created. This display takes all 38 columns in the file window. The optional display shows two columns of filenames (side by side) with extensions, and their attributes. The attribute letter is displayed if set or a dot if cleared.
- Q **Quit** — is the correct way to exit the menu back to the DOS command PROMPT. NOTE: Do not quit or stop operations by pressing <RESET>. This is a very bad practice that can lead to unrecoverable files.

- S Sort** — will sort the file display by: Name, Ext, Date, Size. This is a forward sort which defaults to name. To permanently sort directories or reverse sort them, use SORTDIR from the SpartaDOS Tool Kit.

MKDIR (Make Directory) Command

Purpose This command creates a subdirectory.

Syntax MKDIR [d:]path

Alias MD & CREDIR

Type Internal

Related CHDIR, RMDIR, PATH

Remarks If you do not specify a drive, the default drive is assumed. This function is not supported by the ATARIDOS.SYS driver even though subdirectories are supported by that driver.

Directories (also called subdirectories or folders) are used like file folders to organize your files. They also keep a large storage area fast. In a file cabinet it is much quicker to go to a file folder and search through a few documents, than a pile of all your documents. Computers work the same way. It is much quicker for DOS to go directly to a subdirectory and search through a few files than it is to search through one long file list.

Directory names are stored like filenames but marked with the +S attribute bit. They may not be renamed or deleted in the normal ways in which files are. To rename a subdirectory you must copy all files from inside it to another area, then delete all the files in it, use RMDIR to delete the directory, then create a new name with CREDIR, and copy the files back to it. Otherwise, you can use RENDIR.COM from SpartaDOS Tool Kit.

```
MD TEST
MKDIR 3:>MODEM>TEST
```

The first example creates a subdirectory on the default drive called "TEST". The second example creates a subdirectory on D3: by the name of "TEST" in the subdirectory called MODEM which is in the MAIN directory.

PATH (Set Search Directory) Command

- Purpose** Causes specified directories to be searched for commands before searching the current directory.
- Syntax** PATH [path_string]
- Type** Internal
- Related** CAR, CHDIR, MKDIR, PROMPT, RMDIR
- Remarks** You may specify a list of drives and path names separated by semicolons. Then when you enter a command, SpartaDOS searches the named directories in the sequence you entered them (from path_string) before searching the current directory of the drive that was specified (or implied). The current directory is not changed after the search.

Entering PATH with no parameters causes SpartaDOS to display the current setting of the PATH string.

It is recommended that you include “CAR:” as a device in the search path as this device contains many external commands (such as X, CAR, MENU, DUMP, CHTD, etc.) that you may need. It is also good practice to use “>” or “\” at the start of a device path to force a start at the MAIN directory. The command:

```
PATH A:>;1:>DOS;CAR:
```

sets the search to the root directory of drive A (alias drive D1:), the “DOS” directory of drive 1, and the CARtridge directory.

The PATH command is really just a convenient form of the SET command, for example the above command could also be performed by:

```
SET PATH=A:>;1:>DOS;CAR:
```

The only way to clear the search path to search just the current directory (i.e no search path at all) is the command:

```
SET PATH
```

Where no path has been specified, the system defaults to:

```
PATH CAR:
```

This means search the CAR: device first, then search the current directory. The current directory will always be searched last unless it is included in the path_string. e.g.

```
PATH ;CAR: or PATH :;CAR:
```

The previous examples both mean the same thing; search current directory first, then CAR:, then current directory again. (Remember that current directory is always searched last even if it was already searched.) The stand alone “:” or a space, indicate the current directory.

NOTE: While not required, it is strongly recommended that CAR: always be the first entry in the path string. The programs in this directory are called often. If any other devices are listed first, they will always be checked before the CAR: device, slowing system response considerably.

For more information, see the SET command in chapter 5.

PAUSE Command

- Purpose** Suspends system processing and displays the message “Press <RETURN> to continue”.
- Syntax** PAUSE
- Type** Internal
- Related** DIR, TYPE
- Remarks** You can insert PAUSE commands within a batch file to provide the opportunity to change diskettes between commands or to step through a process, giving you time to read instructions, etc.

To resume execution of the batch file, press the <RETURN> key.

NOTE: It is very dangerous to change diskettes (during a PAUSE) on the drive from which the batch file was running. If using PAUSE to change diskettes, run the batch file from a RAMDISK or another drive which will not be changing.

PEEK Command

Purpose To examine a memory location or perform a HEX conversion.

Syntax PEEK [\$]location

Type Internal

Related POKE

Remarks PEEK allows examination of a memory location from the command processor. It is also useful as a quick DEC to HEX or HEX to DEC convertor. (DEC means decimal or base 10; HEX means hexadecimal or base 16.) PEEK returns the dec and hex value of the location entered, the contents of the location in both dec and hex, the dec and hex value of the memory word stored in the location and location+1, and the ATASCII character representing the value of the location.

It is a good idea to PEEK a location before POKEing it if you are not sure what you are doing. You can usually recover by POKEing the old value back in (unless the computer has crashed).

POKE Command

- Purpose** To change the contents of a memory location.
- Syntax** POKE [\$]location [\$]value
- Type** Internal
- Related** PEEK
- Remarks** POKE allows you to change memory locations from the command processor which can be useful in batch files and other applications. It is very easy to crash the system with this one if you do not understand what you are doing. A few examples of POKE locations and useful values follow:
- POKE 77 (attract) 0=attract mode off for a few minutes
 - POKE 82 (lmargin) n=number from 0 to 39 for left margin
 - POKE 83 (rmargin) n=number from 0 to 39 for right margin
 - POKE 559 (sdmctl) 0=screen off, 34=screen back on
 - POKE 710 (color2) 0=black, 53=red, 148=blue
 - POKE 730 (keyrep) 1=hyper, 3=fast, 5=normal (XL/XE only)
 - POKE 731 (noclik) 0=normal, 1=speaker off (XL/XE only)
 - POKE 752 (crsinh) 0=cursor off, 1=cursor on
 - POKE 702 (shflok) 0=lower case, 64=upper case
 - POKE 65 (soundr) 0=SIO sound off, 1=SIO sound on
- A good memory map will give many more and is a must for programming the Atari.

PROMPT (Set System Prompt) Command

Purpose Change the system prompt.

Syntax PROMPT [prompt_string]

Type Internal

Related PATH

Remarks The text in prompt_string is taken by SpartaDOS to be the new system prompt. Special meta-strings can be imbedded in the text in the form “\$c” where ‘c’ is one of the following characters:

L print current drive letter ('A' through 'I')

N print current drive number ('1' through '9')

P print path on current drive

D print current date

T print current time

R print an EOL character (advance to next line)

NOTE: “P” will cause the current drive to be read every time <RETURN> is pressed to detect a disk change. This should be taken out of the path before parking a hard drive, since the “P” will read the disk and unpark the drive. A batch file changing the prompt, then parking the drive, would be very useful for this purpose.

If no parameter is specified, then the current prompt_string is displayed.

For example the command:

```
PROMPT $L:$P>
```

will display a prompt in the form:

```
B:>DOS>
```

assuming the current drive is 2 and the current path is “DOS”. Also, the ‘_’ character will display as a space rather an underline. (Thus a prompt can end in a space.)

The PROMPT command is really just a convenient form of the SET command, for example the above command could be performed by:

```
SET PROMPT=$L:$P>
```

just as easily. The default value of the “prompt” variable is “D\$N:” which displays the same prompt as older SpartaDOS versions. If the “PROMPT” variable is not defined, SpartaDOS X will prompt with a ‘>’ character — the only way to clear the “PROMPT” variable is with the command:

```
SET PROMPT
```

You can not use non-inverse lower case letters in the prompt because the command processor automatically converts all lower case characters to upper case before processing. Inverse and cursor control keys (preceded by the escape key) may be used in the prompt.

When using the ‘\$P’ meta-string in the prompt, the default drive will be read each time the prompt is printed. This will cause an error to be printed within the prompt if there is no disk or a bad disk in the default drive, or if the disk is of a format unrecognized by SpartaDOS X. (To use Atari DOS format disks with SpartaDOS X you must install the ATARIDOS.SYS driver).

Using the '\$P' meta-string in the prompt can also cause problems when attempting to park a hard drive, since the drive will be "unparked" to read the path when the prompt is printed. The solution to this is to set the environment variable PROMPT to a value not containing '\$P'. Since drives are usually parked only when you are through using the computer, you may simply clear the PROMPT variable with a

```
SET PROMPT
```

before parking the drive. You could set up a batch file to clear the prompt variable and then park the drive to simplify this operation.

RENAME Command

Purpose This command allows you to change the name of one or more files.

Syntax RENAME [d:][path]fname[.ext] fname[.ext]

Alias REN

Type Internal

Related MENU

Remarks Wildcards may be used in both filespecs. A device and path may only be specified on the first filename (the old name filespec). Filenames must be specified for both source and destination names, otherwise an error will occur. The rules for wildcarding are the same as for the COPY command. Here are a few examples:

```
RENAME *.BAK *.DOC
```

The above command changes all the extensions to “DOC” of files that previously had extensions of “BAK”.

```
RENAME AC*.* *.XX
```

This command changes the extension of all files beginning with “AC” to “XX”.

NOTE: Rename your files with caution. There is no check for existing filenames. Careless renaming may result in more than one file with the same name. The only way to separate these files is with a sector editor like DiskRx (from SpartaDOS Tool Kit) or through tedious use of the ERASE and UNERASE commands. To use this second method, ERASE the duplicate files, then UNERASE them. Answer 'Y' to only one of them and 'N' to the rest. Rename this file to something different and repeat the this UNERASE, RENAME cycle until all of the files are restored.

RMDIR (Remove Directory) Command

Purpose This command deletes an empty subdirectory from the specified drive.

Syntax RMDIR [d:]path

Alias RD & DELDIR

Type Internal

Related CHDIR, MKDIR, PATH

Remarks The directory must be empty before it can be removed. The last directory name in the path is the directory to be removed. This function is not supported by the ATARIDOS.SYS driver even though subdirectories are supported by that driver.

```
RD TEST
DELDIR 3:>MODEM>TEST
```

The first example removes a subdirectory on the default drive by the name of "TEST". An error will occur if the directory has files in it. The second example removes a subdirectory on drive D3: by the name of "TEST" in the subdirectory called MODEM which is under the MAIN directory.

For related information see the CHDIR, MKDIR, and PATH commands.

Note: If a file has been opened for write or update but not properly closed (usually by hitting reset or losing power while it is opened) its entry in the directory will not be removed, although it may not show in a listing. A subdirectory containing a "phantom" entry of this type can not be deleted. CLEANUP or DISKRX from the SpartaDOS Tool Kit can be used to mark such an entry as deleted and not open so that the directory may be removed. The status byte of the directory entry will have bit 7 and bit 3 set. These should be cleared and bit 4 set. It is possible that some sectors have been allocated for this file. These should be de-allocated in the bit map.

RS232 (Load RS232 Driver) Command

Purpose This command loads the RS232 handler from a P:R: Connection or the Atari 850 interface.

Syntax RS232

Type External — on device CAR:

Remarks You need to use this command prior to using a P:R: Connection or Atari 850 interface **unless** the program you are going to use does this automatically. Try your program without RS232 first. You should hear a beep on your monitor (TV) speaker if the handler loads. If not and an error occurs, type this command and run your program again.

Avoid loading the RS232 handler more than once. Your system may crash if you load several copies of the RS232 handler into memory, since MEMLO is raised each time.

SAVE Command

Purpose This command saves binary data from memory to disk.

Syntax SAVE [d:][path]fname[.ext] [\$]address [\$]address

Type Internal

Related LOAD

Remarks Addresses are assumed to be decimal unless preceded by a '\$' (which indicates HEX). This is useful when used in conjunction with the LOAD command for desegmenting MAC/65 files or to save a snapshot of memory for debugging purposes.

SET Command

Purpose To display the values of all environment variables, and optionally to set an environment variable to a specified value.

Syntax SET [var[=env_string]]

Type Internal

Remarks Environment variables are global strings that can be used for one program to communicate to another with. For example the “CAR” variable tells the CAR command where the memory-save file is. There are three forms of this command. For example the command:

```
SET
```

displays the contents of all environment variables, and:

```
SET CAR=A:CAR.SAV
```

sets the variable “CAR” to the value “A:CAR.SAV”. The command:

```
SET CAR
```

deletes the environment variable “CAR” from the system. (This will cause the CAR command to not use a memory-save file.)

SWAP (Swap Drives) Command

Purpose This command allows you to swap (re-map) your drive configuration.

Syntax SWAP [d,d]

Type Internal

Remarks SWAP, without any parameters, will display the drive map list showing drives 1 through 9. The default is 1=1, 2=2, etc.

For example, to interchange drives 1 and 9, type the following command:

```
SWAP 1,9      (or)
SWAP A,I
```

The order is not important, so 1,9 is equivalent to 9,1.

The drives will stay mapped that way unless remapped or a COLD start occurs. Note that you may use letters or numbers to reference a drive and that no colon (":") follows the drive specifier.

SWAP works in *addition* to Multi I/O drive remapping, so take that into consideration when using the Multi I/O. It is very easy to lose track of which floppy, RAMDISK, or hard drive partition is at which logical drive.

TD (Time/Date Display) Command

- Purpose** This command allows you to turn on and off a time/date display line on top of your screen.
- Syntax** TD ON|OFF
- Type** External — on device CAR:
- Related** CHTD, DATE, TIME
- Remarks** This command is much like the KEY command in the way it links into your computer.
- You must have either the JIFFY.SYS or CLOCK.SYS driver installed before you can use this command. It calls one of these drivers directly (through the I_GETTD symbol) and will not load without one of these drivers. (These drivers are loaded by default unless you are using your own CONFIG.SYS file.)
- NOTE: TD ON may be incompatible with some programs. If you are having problems with a program, try TD OFF, or do not install it at all.

TIME Command

Purpose This command displays the current time and allows you to set the time.

Syntax TIME

Type Internal

Related CHTD, DATE, TD

Remarks This command produces the following output

```
Current time is 14:34:30
Enter new time:
```

You may enter the new time or press <RETURN> if you don't want to change it. Enter the time in the format "hh:mm:ss" where "hh" is the hour (24 hour clock), "mm" is the minutes, and "ss" is the seconds. (SpartaDOS 3.2 uses AM/PM — SpartaDOS X uses a 24 hour clock.)

This command will produce meaningless results if you do not have a clock driver installed in your system. The two clock drivers are "CLOCK.SYS" and "JIFFY.SYS" — the first being for the R-Time 8 and the second uses the jiffy counter to keep its time. By default, one of these drivers will be installed when you boot, but this can be overridden by creating a custom "CONFIG.SYS" file and not including these drivers in the configuration.

TYPE Command

- Purpose** This command displays the contents of a specified file.
- Syntax** TYPE [+A|H|P|S] [-A|H|P|S] [d:][path]fname[.ext] [/P]
- Type** Internal
- Related** COPY, DUMP, MENU, PAUSE
- Remarks** You may display any file and are not limited to a maximum line length (as was the case with SpartaDOS 3.2). Press <CTRL-1> to stop and start the display. You may specify attributes as in the DIR command — the default attributes are “-HS”. (See the DIR command for a description of the attributes.)
- If you include a “/P” parameter, the TYPE command will wait for a keypress after each 23 lines of text.

UNERASE Command

Purpose This command restores files previously erased (if possible).

Syntax UNERASE [d:][path]fname[.ext]

Type External — on device CAR:

Related ERASE

Remarks Wildcards are permitted. You will be asked (for each recoverable file matching the filespec you entered) if you wish it restored or not. If you think that erased files exist in your directory that were not seen by UNERASE, they were not recoverable for one of two reasons:

- 1) The file's directory entry has been allocated to another file which was copied to the directory after the original file was ERASEd.
- 2) A sector of the file has been allocated to another file since the original file was ERASEd.

VERIFY Command

- Purpose** To turn write verify on or off.
- Syntax** VERIFY ON/OFF
- Type** Internal
- Remarks** When ON, SpartaDOS performs a verify operation following each disk write operation, to verify that the data just written can be read without error. This only applies to floppy drives. Because of the extra time required to perform the verification, the system runs much slower when programs write data to disk. The default is OFF — this command is typically used when you are having floppy drive problems.

X Command

- Purpose** Execute a program which requires that no cartridges are installed (such as DISKRX, EXPRESS, most binary files, etc.)
- Syntax** X [d:][path]fname[.ext] [parameters]
- Type** External — on device CAR:
- Remarks** There are four possible environments a command can run under. They are:
- 1 with internal BASIC present (via BASIC command)
 - 2 with a CARtridge present (via CAR command)
 - 3 with the SpartaDOS X library enabled (just type the program or command name)
 - 4 with no cartridges present (via the X command)

The first three modes use the SpartaDOS X library to perform various DOS functions, including loading and running the command. The fourth mode, however, cannot use the library without moving or disabling the screen! Thus, the following features are disabled when commands are run with this command:

- The search path is not used — you must specify the exact location of the file if it is not in the current directory of the current drive.
- The mini-buffers are not used — single byte get/put will be *very* slow (this is extremely rare since most programs that use single get byte and put byte are in BASIC or use a cartridge)
- Since the library is disabled, only standard binary files are loadable — SpartaDOS external commands such as FIND and MENU can not be run.
- I/O redirection is severely hampered because it must use the library. In doing this the screen will flicker rapidly.

The general rule of thumb is: “If a program does not work with a cartridge installed, prefix the command with the X command, otherwise, just type the command.”

The Command Processor — Commands

X.COM remains resident in memory while the program called is running, so MEMLO will be slightly higher until the program is exited to the command processor.

Executing a cold start (a jump to \$E477) while using X.COM will disable the SpartaDOS X cartridge and the external cartridge, if present.

The Command Processor — Advanced Features

The SpartaDOS X command processor has been greatly enhanced with more sophisticated batch files, command line I/O redirection, user definable prompts, command search paths, and more.

This chapter discusses these features and gives many examples. Most of these features are either new to SpartaDOS or have been greatly enhanced over prior versions of SpartaDOS.

Batch Files Batch files are simply a list of SpartaDOS commands that may be fed to the command processor from a text file. Parameters may be passed to the batch file by including them on the command line following the batch file name. The syntax is

```
-fname [parm1 parm2 ... parm9]
```

The filename (“fname”) is assumed to have an extension of “.BAT” although this assumption may be overridden by including an extension on the command line. The parameters (“parm”) are optional.

The following is an example of a batch file which accepts two files on input and creates an output file containing the contents of both source files (we will call this file “TEST.BAT”).

```
COPY %1 %3  
COPY %2 %3/A
```

Now, the command

```
-TEST FILE1 FILE2 OUTPUT
```


will concatenate the files “FILE1” and “FILE2” to the file “OUTPUT”.

You may pass up to 9 parameters (numbered “%1” to “%9”) to a batch file. The parameter “%0” is the name of the batch file (in the above case, this would be “TEST”). The ‘%’ parameters may appear anywhere in the batch file and may be surrounded in text (i.e. no spaces need precede the ‘%’ character).

The batch file parameters are automatically saved in environment variables “_x1” where ‘x’ is the parameter number. Due to the command processor’s non-resident nature, the parameters need to be saved somewhere — environment variables are a perfect place. This also means that the number and size of parameters is limited to a total of 256 characters less overhead (the “_x1=” string) and space used by other environment variables.

Default Batch File

When the command processor is first entered, it tries to run a batch file called “AUTOEXEC.BAT”. You should put any system setup commands that you may need in this file.

The variable “BATCH” will be read by the command processor just before it prints its prompt. If this variable exists and contains a filename, this file will be executed as a batch file. As soon as the command processor reads the variable, it will delete it from the list of environment variables. This is how the system passes the “AUTOEXEC” filename to the command processor when it is loaded for the first time. The variable BATCH, then, can be used to cause a batch file with a different name to be executed on boot by using the line

```
SET BATCH=d:filename.BAT
```

in CONFIG.SYS.

I/O Redirection

You may redirect the standard input and output of SpartaDOS X commands on the command line. With SpartaDOS 3.2, redirection was done using batch files (for input redirection) and the PRINT command (for output redirection). SpartaDOS X implements I/O redirection in a totally different manner. Batch files are no longer considered as “input redirection” — they are *only* read by the command processor and are **not** system wide (i.e. you may not feed input to BASIC through a batch file). The PRINT command has been eliminated.

With SpartaDOS X, you may divert output of a single command by including “>>d:fname” on the command line. Similarly, input redirection is accomplished by including a “<<d:fname” on the command line. For example, the command

```
DIR >>PRN:
```

redirects the output from the DIR command to the printer (the directory listing will not appear on the screen). An alternate way to copy a file would be

```
TYPE fname >>dest
```

This will be slower than the copy command and will not copy the date to the new file.

```
BASIC <<AUTOGO
```

will run the BASIC program “START.BAS” if the text file “AUTOGO” contains the line

```
RUN "D:START.BAS"
```

As an example of output redirection, the following batch file will allow paged viewing of the output of any program by redirecting it to a temporary file, then TYPEing the temporary file with the pause option:

```
%1 %2 %3 %4 %5 %6 %7 %8 %9 >>TEMP
TYPE TEMP /P
PAUSE
DEL TEMP
```

For example, if you named the above batch file MORE.BAT and wanted to read the directions printed by the ARC program, use

```
-MORE ARC
```

Search Path

Whenever a command is given to the command processor without a drive and/or path being specified, a check is made to see if it is an internal command (such as ERASE). If not, the list of installed external commands (such as TD or KEY after they have been run once) is searched. If the command is not found, then a check is made to see if the environment variable PATH exists. If it does, all of the devices and/or paths named in the variable are checked for the command in the order specified. If the command is still not found, the default directory is searched. The PATH variable provides a high degree of flexibility and power, allowing you to keep often used utilities out of the way in subdirectories. This is particularly useful if you own a hard drive, since the main directory of D1: can get very cluttered.

You can examine the PATH variable by typing

```
PATH
```

with no parameters at the CP prompt. The default PATH is

```
CAR:
```

The search path may be changed by typing

```
PATH path1;path2;...;pathn
```

at the CP prompt. Each device and/or path specified must be separated from the others with a semicolon (;). It is a good idea to leave CAR: as the first entry, since many often used commands are in this device and it can be accessed much more rapidly than the others. Order is important, since the entries will be searched one after the other. For example,

```
PATH CAR:;A:\DOS\;A:\TOOLKIT\;D9:>;A:>;>
```

will search the CAR: device, the directory DOS on D1:, the directory TOOLKIT on D1:, the RAMDISK at D9:, the main directory of D1:, the main directory of the default drive, and then the current directory of the default drive.

If a path is specified on the command line, this search path will *not* be used.

Programming with SpartaDOS X

SpartaDOS X Functions from BASIC

Many features of SpartaDOS may be accessed in BASIC, Action!, machine language, and other programming environments. The following is a list of common BASIC functions and XIO statements that allow the programmer to accomplish a variety of tasks. Conversion to other languages should not be difficult; refer to the reference material for that language for details.

In this list, *IOCB* refers to an Input/Output Control Block (or channel) number from 0 to 7. IOCB #0 is used by the Atari operating system for the screen editor, so it should normally not be used. An *Atari DOS disk* is one initialized in standard Atari DOS 2 format, whether in single, enhanced (dual), or double density, as produced by Atari DOS 2.0S and 2.5, MYDOS, other DOS 2 clones, and the SpartaDOS X Formatter when used in Atari DOS mode. *d.*, *path*, and *fname.ext* refer to any legal SpartaDOS X device identifier, pathname, and filename with extension as defined in chapter 4.

Notes on the Default Drive *Please remember that D: from BASIC or another language refers to the default drive, not necessarily drive #1. From the command processor, D: refers to drive #4. With most other DOS types including earlier versions of SpartaDOS, D: represents D1:.*

```
OPEN #1, 4, 0, "D:TEST.TXT"
```

will open the file TEST.TXT on the *default drive*, not necessarily drive #1, for read under SpartaDOS X.

SpartaDOS X Reference Manual

Accessing the “Kernel” Through CIO

The D: device available through the CIO with SpartaDOS X is not just the disk drive handler; it is the handler for the SpartaDOS “kernel”. Any “kernel” device may be accessed through the CIO from any application by preceding its name with D. For example,

```
OPEN #3,8,0,"DPRN:"
```

will open the printer for output. This also means that D4:, DD:, DD4:, DDD:, DDSK4:, and DDSKD: all refer to drive #4. When referring to a device other than a disk drive or the CAR: device, the fname.ext part of the syntax is ignored. If this confuses you just ignore it and use D1: - D9:, E:, P:, R:, and so on as you would with any other DOS.

Open File

Purpose

To open a disk file for access through SpartaDOS X.

Syntax

```
OPEN #IOCB,aux1,aux2,"Dd:[path]fname.ext"
```

Remarks

This command opens a disk file through SpartaDOS X. **Aux1** is the mode (output, input, update, directory, etc.) in which the file will be opened. The following is a list of legal values for **aux1**. Unless otherwise noted, **aux2** should be 0.

- 4 Open the file in read only mode.
- 6 Open a formatted directory. Provides a directory listing as do the DIR and DIRS commands from the command processor. **Aux2** is used to determine the style of the directory. If **aux2** is 0, standard DIRS format will be used. If **aux2** is 128, the long DIR format, including size in bytes, date, and time, will be used.
- 8 Open the file in write only mode.

- 9 Open the file in append mode. Data will be written to the end of an existing file. If the file does not exist it will be created.

- 12 Open the file in update mode. This mode allows both reading from and writing to a file. Note: On a SpartaDOS format disk it is possible to position and/or write past the end of a file while in update mode.

An Example

This short BASIC program will read the formatted directory of a disk in drive #1 in long format and print it to the screen:

```

10 DIM ENTRY$(40)
20 OPEN #1,6,128,"D1:*.*)"
30 REM The TRAP will cause the program to jump to line
40 REM 80 when the end of the directory is reached.
50 TRAP 80
60 INPUT #1,ENTRY$:PRINT ENTRY$
70 GOTO 60
80 CLOSE #1

```

Accessing the Raw Directory

Setting bit 4 of **aux1** puts the OPEN in raw or unformatted directory mode. This allows you to read from and/or write to SpartaDOS directories as if they were normal data files. Although this is much faster than reading a formatted directory, there is no easier way to trash a disk and make it unusable than to make a mistake in the raw directory. Unless you feel confident about what you are doing and are using a disk you don't mind losing, stay away from the raw directories! This mode *will* work with Atari DOS disks if the ATARIDOS.SYS driver is installed. The driver translates the Atari directory format into SpartaDOS format and back.

Scan Mode

Adding 64 to **aux1** will place the OPEN in attribute scan mode. **Aux2** is used to determine the attributes desired. If a long directory is wanted in scan mode, then 128 should be added to **aux1** instead of to **aux2**.

To determine the file attributes to be scanned, the following values should be added to **aux2**, assuming an initial value of 0:

SpartaDOS X Reference Manual

Protected	add 1	Unprotected	add 16
Hidden	add 2	Not hidden	add 32
Archived	add 4	Not archived	add 64
Subdirectory	add 8	Not a subdirectory	add 128

Only those files that fit the requested description will be referenced. A value of 0 in **aux2** will ignore all attributes, even "Hidden".

For example, to get a long format directory of only the hidden files in the BASIC example on the previous page, simply substitute the following line:

```
20 OPEN #1,6+64+128,2,"D1:*.*"
```

For a short directory listing without subdirectories, use

```
20 OPEN #1,6+64,128,"D1:*.*"
```

and, finally, for a long listing of the unhidden, protected entries that end with a .COM extender, use

```
20 OPEN #1,6+64+128,1+32,"D1:*.COM"
```

It is possible to select contradictory conditions (such as 1+16, protected and not protected) for each of the attributes. This will not produce an error but, since no directory entry can match both conditions, will always select no files.

Rename File(s) (RENAME)

Purpose To change the name of a file or group of files.

Syntax XIO 32, #IOCB,0,0,"Dd:[path]fname1.ext fname2.ext"

Remarks The name of the file or names of the files specified by **fname1.ext** will be changed to **fname2.ext**, exactly as with the RENAME command from the command processor. The IOCB selected should be closed for this operation. Wildcards may be used in both file name specifications. **WARNING!** SpartaDOS X has an extremely powerful RENAME function. It is possible to give two or more files on one disk the same name. It will then be impossible to refer to one file without referring to the other(s). For a few verbose ways to recover from duplicate file names, refer to the RENAME command remarks in chapter 4.

Erase File(s) (ERASE)

Purpose To remove unwanted files from a disk.

Syntax XIO 33, #IOCB,0,0,"Dd:[path]fname.ext"

Remarks The file or files specified will be erased from the disk. The IOCB selected should be closed for this operation. Wildcards may be used. While it is possible to recover erased files in some instances (see the command UNERASE in chapter 4), it is important to be very careful with this command.

Protect File(s) (ATR +P)

Purpose To prevent a file or files from being changed or erased.

Syntax XIO 35, #IOCB,0,0,"Dd:[path]fname.ext"

Remarks This will allow the specified files to be opened in read mode only. Wildcards may be used. The IOCB should be closed. Protected files may not be erased, changed, overwritten, or renamed.

Unprotect File(s) (ATR -P)

- Purpose** To allow files previously protected to be changed or erased.
- Syntax** XIO 36, #IOCB,0,0,“Dd:[path]fname.ext”
- Remarks** This removes the protected status of selected files previously protected with the ATR command or the Protect Files XIO command above. The file or files may now be erased, renamed, or changed. Wildcards may be used. The IOCB should be closed.

Set File Position - POINT

- Purpose** To allow direct access to specific points within a disk file (or past the end of a file if necessary).
- Syntax** X=POS
Y=0 (see text)
POINT #IOCB,X,Y
- or*
- A=INT(POS/65536)
B=INT((POS-A*65536)/256)
C=POS-A*65536-B*256
POKE 844+IOCB*16,C
POKE 845+IOCB*16,B
POKE 846+IOCB*16,A
XIO 37, #IOCB,aux1,aux2,“Dd:”

Remarks

Unlike Atari DOS and compatibles, which use an absolute physical disk position (sector and offset into sector) for the NOTE and POINT functions, SpartaDOS X uses a relative position within the file. POS is the desired offset into the currently open file. For example, if POS was 612, the next GET from the file would get the 613th byte of the file. This value will refer to the same position in the file even if the file is physically moved to another disk. The file must be open for this operation.

Because of a limitation in Atari BASIC, BASIC XL, and BASIC XE, the first method shown, using the POINT command, will only work with positions up to and including 32767. If a value greater than 32767 is given a BASIC error will occur. To POINT to a greater location with these languages (and possibly others) it is necessary to use the second method. The POINT command is bypassed by poking the three byte file position directly into the IOCB registers and executing the XIO. Aux1 and aux2 must be the values used when the file was opened.

Other languages, such as Action! and Turbo BASIC XL, have no such limitation on the POINT command, allowing it to be used instead of the lengthy XIO method. In this case, use the following format:

```
Y=INT(POS/65536)
X=POS-Y*65536
POINT #IOCB,X,Y
```

If you are a user of an earlier version of SpartaDOS, you should notice that NOTE and POINT now work the same way with Atari DOS disks as they do with SpartaDOS disks. POINT will *not* use sector number and offset regardless of disk format.

Using NOTE and POINT with SpartaDOS X and an Atari DOS disk may prove to be time consuming since, to determine the relative offset into the file, it is necessary to read the file from the beginning every time a POINT is used. This also causes segmented binary files to take much longer to load from Atari DOS disks than from SpartaDOS disks. NOTE and POINT tables created by other DOS types (including earlier versions of SpartaDOS for files on *Atari DOS disks*) will, of course, no longer be valid.

Sparse Files

On a SpartaDOS diskette, it is possible to point past the end of a file opened in append mode. When data is placed in a file past the end, the file is given the new length, but no physical sectors are used for the space between the old and the new data. In the sector map of the file, the unallocated sectors are represented by a sector number of 0. Should you at any time write to a position in this gap, a sector will be allocated. This gap may not be read, and a file containing gaps may not be copied. An error will occur if either of these is attempted.

Get Current File Position - NOTE

- Purpose** To determine the current position within a file.
- Syntax** NOTE #IOCB, X, Y
POS=X+65536*Y
- Remarks** This will return the relative current position within the currently open file; i. e., the offset into the current file. This will *not* return sector number and offset into the sector, regardless of disk format. The file must be opened for this operation.
- For users of SpartaDOS 2.x and 3.x, you may be interested to learn that this method works with those versions. The XIO 38 command described in the SpartaDOS Construction Set manual will still work but is unnecessary.

Get File Length

Purpose	To determine the length of the currently open file.
Syntax	XIO 39, #IOCB,aux1,aux2,"Dd:" A=PEEK(844+IOCB*16) B=PEEK(845+IOCB*16) C=PEEK(846+IOCB*16) LNGTH=A+B*256+C*65536
Remarks	This will return the length of the currently open file. IOCB , aux1 , and aux2 should be the same values used when opening the file.

Load a Binary File (LOAD)

Purpose	To load and execute a binary file from another program.
Syntax	XIO 40, #IOCB,4,0,"Dd:[path]fname.ext"
Remarks	This command will load a binary file and execute it using the INIT and/or RUN vectors. The IOCB should be closed. Loading a binary file from an AtariDOS disk will take much longer than loading the same file from a SpartaDOS format disk.

Create a Directory (MKDIR)

Purpose	To create a new subdirectory.
Syntax	XIO 42, #IOCB,0,0,"Dd:[path]newdir"
Remarks	The directory " newdir " is the directory that will be created. Any path before this must be valid. For example, if <pre>XIO 42, #1, 0, 0, "D1:LARRY>MOE>CURLY>SHEMP"</pre> is used, then the path "LARRY>MOE>CURLY>" must already exist from the current directory, and the directory "SHEMP" will be created.

SpartaDOS X Reference Manual

The IOCB should be closed for this operation. This will ONLY work for SpartaDOS format disks.

Delete a Directory (RMDIR)

Purpose To remove an existing directory.

Syntax XIO 43, #IOCB,0,0,“Dd:[path]olddir”

Remarks The directory **olddir** will be deleted. A directory must be empty to be deleted. The rules regarding path and IOCB status defined in XIO 42 apply here.

Change Current Directory (CHDIR)

Purpose To change the current working directory of a disk.

Syntax XIO 44, #IOCB,0,0,“Dd:path”

Remarks This will change the directory that is used when the specified drive is accessed without reference to a specific directory. The rules regarding path and IOCB status defined in XIO 42 apply here.

Set Boot File (BOOT)

Purpose To establish the file that will be loaded when the computer is initialized when SpartaDOS X is not used.

Syntax XIO 45, #IOCB,0,0,“Dd:[path]fname.ext”

Remarks This will cause the specified file to load when the computer is turned on or cold started and the SpartaDOS X cartridge is not used. With earlier versions of SpartaDOS, the primary use of this was to cause the *.DOS file to be booted. With SpartaDOS X, the uses of this command are limited. The IOCB should be closed, and a SpartaDOS format disk must be used. **Note:** BOOT will not work with all binary files. There are many specific rules that must be followed when loading a file without DOS. The primary purpose of this command is to load a DOS module.

Set Attributes (ATR)

Purpose To manipulate the protected, hidden, and archived status of files.

Syntax XIO 49, #IOCB,aux1,aux2,“Dd:fname.ext”

Remarks This is used to modify the attributes of a file or files. Wildcards are allowed in **fname.ext**. **Aux1** is used to select which attributes are to be changed and whether they will be set or cleared. **Aux2** is used to determine which files are to be affected.

To perform the desired attribute modification, add the following values to **aux1**, assuming an initial value of zero:

Protect	add 1	Unprotect	add 16
Hide	add 2	Unhide	add 32
Set archive	add 4	Clear archive	add 64

Aux2 is used exactly as with the scan mode of the open statement. It will select the files to be affected by current attribute status. These values should be added to **aux2**, starting with a base value of 0:

Protected	add 1	Unprotected	add 16
Hidden	add 2	Not hidden	add 32
Archived	add 4	Not archived	add 64
Subdirectory	add 8	Not a subdirectory	add 128

For example, to hide all of the files on drive #1 with a .BAK extender, use

```
XIO 49, #1, 2, 0, "D1:* .BAK"
```

To protect and set the archive bit for all of the hidden files on drive #1, use

```
XIO 49, #1, 1+4, 2, "D1:* .*"
```

and to unhide and unprotect all the hidden files with a *.BAK extender on drive #1, use

```
XIO 49, #1, 16+32, 2, "D1:* .BAK"
```

The IOCB should be closed for this operation.

Format a Disk (FORMAT)

- | | |
|----------------|--|
| Purpose | To initialize a disk, setting up the appropriate track, sector, and directory data. |
| Syntax | XIO 254, #IOCB,0,0,“Dd:” |
| Remarks | The Dd: specified is irrelevant, since this command will bring up the SpartaDOS X disk formatter menu. From this menu disk number, format, size, skew, etc. may be selected. Once the formatter is exited with the <ESC> key, control will be returned to the program. This allows disks of all types to be formatted from within any program. The IOCB should be closed for this operation. WARNING! Formatting a disk will destroy all existing data on the disk. Hiding or protecting a file will not save it from being destroyed during a disk format. |

The next two commands are not available through XIO calls. They must be accessed directly through the CIO. An assembly language listing of a routine to access these will follow, along with a BASIC program that demonstrates its use.

Get Disk Information (CHKDSK)

Purpose To read information about a disk

CIO Data

iccom = 47
 icbal = low byte of 'Dd:' address
 icbah = high byte of 'Dd:' address
 icbll = low byte of buffer address
 icblh = high byte of buffer address

CIO Output Results

buffer = results of CHKDSK operation (17 bytes)
 +0 = version number of disk, 0 if Atari DOS format
 +1 = number of bytes per sector, 0 if 256
 +2 = total number of sectors on disk (2 bytes)
 +4 = Number of free sectors on disk (2 bytes)
 +6 = volume name, always "AtariDOS" for Atari DOS
 format disks (8 bytes)
 +14 = volume sequence number, 0 if Atari DOS format
 +15 = volume random number, 0 if Atari DOS format

Get Current Directory Path (CHDIR)

rpose To get the path from the root directory to the current directory of a drive

CIO Data

iccom = 48

icbal = low byte of 'Dd:[path]' address
icbah = high byte of 'Dd:[path]' address
icbll = low byte of buffer address
icblh = high byte of buffer address

An Example

The following is a short BASIC program demonstrating the use of the last two CIO calls. It is followed by an assembly language listing of the code contained in the DATA statements and later in the string CIO\$.

```
10 DIM CIO$(32),BUFFER$(64),DRIVE$(4),CHKDSK(17)
20 DRIVE$="D1: ":DRIVE$(4)=CHR$(155)
30 RESTORE 50
40 FOR X=1 TO 32:READ Y:CIO$(X)=CHR$(Y):NEXT X
50 DATA 104,104,104,10,10,10,10,170,104,104,157,66,3
60 DATA 104,157,69,3,104,157,68,3,104,157,73,3,104,157
70 DATA 72,3,76,86,228
80 REM MAIN LOOP
90 BUFFER$(1)=CHR$(0):BUFFER$(64)=CHR$(0)
100 BUFFER$(2)=BUFFER$
110 ?:"CIO Call Demonstrator"
120 ?:"1 -> CHKDSK"
130 ?:"2 -> Path to current directory"
140 INPUT CHOICE
150 IF CHOICE<>1 AND CHOICE<>2 THEN GOTO 120
160 ICCOM=CHOICE+46
170 ?:"Which drive";:INPUT D
180 D=INT(D):IF D<1 OR D>9 THEN GOTO 170
190 DRIVE$(2,2)=STR$(D):IOCB=1
200 X=USR(ADR(CIO$),IOCB,ICCOM,ADR(DRIVE$),
ADR(BUFFER$))
210 IF CHOICE=1 THEN GOTO 270
220 IF BUFFER$(1,1)=CHR$(0) THEN ?"Root directory":
GOTO 80
230 FOR X=1 TO LEN(BUFFER$)
240 IF BUFFER$(X,X)=CHR$(0) THEN BUFFER$(X)=">":
BUFFER$=BUFFER$(1,X):POP:GOTO 260
250 NEXT X
260 ? BUFFER$:GOTO 80
270 FOR X=1 TO 17:Y=ASC(BUFFER$(X,X)):CHKDSK(X-1)=Y:
```

Programming With SpartaDOS X

```
NEXT X
280 ?"      Volume: "; BUFFER$(7,14)
290 ?"Bytes/sector: ";
300 IF CHKDSK(1)=0 THEN CHKDSK(1)=256
310 ? CHKDSK(1)
320 ?" Total bytes: ";
330 ? CHKDSK(1)*(CHKDSK(2)+256*CHKDSK(3))
340 ?"  Bytes free: ";
350 ? CHKDSK(1)*(CHKDSK(4)+256*CHKDSK(5))
360 GOTO 80
```

```
;origin is arbitrary since it will be in a string
ciouv .equ $E456
iccom .equ $0342
icbal .equ $0344
icbah .equ $0345
icbll .equ $0348
icblh .equ $0349
      *=$5000          ; or whatever
      pla             ; number of arguments.
      pla             ; should be 0
      pla             ; iocb channel number
      asl a           ; multiply by 16 for
      asl a           ; proper IOCB form
      asl a
      asl a
      tax             ; in x where it belongs
      pla             ; 0 again
      pla             ; command number
      sta iccom,x
      pla             ; address of "Dx:" string
      sta icbah,x
      pla
      sta icbal,x
      pla             ; buffer address
      sta icblh,x
      pla
      sta icbll,x
      jmp ciouv      ; all done.  Jump CIO
```


SpartaDOS User Accessible Data Table

Several SpartaDOS variables have been made available to programmers to allow easy access to the command line for applications and utilities. This data table is referred to as COMTAB and is pointed to by the OS variable DOSVEC at memory location 10 (\$0A). An assembly language example will follow as an aid. This table is valid with all versions of SpartaDOS except where noted. Locations COMTAB, ZCRNAME, BUFOFF, COMFNAM, and LBUF are also supported by OS/A+ and DOS XL.

DECOUT COMTAB-19

SpartaDOS X only. Contains the right justified, space padded output of the "misc_convdc" routine, an ASCII string representation of the three byte number at DIVEND (see Page Seven "Kernel" Values). (8 bytes)

LSIO COMTAB-10

This is a pointer to the SpartaDOS high speed SIO routine. You can use the address contained here instead of \$E459, the OS SIOV, to perform high speed sector I/O with your programs.

DIVEND COMTAB-6

SpartaDOS X only. A three byte number here will be converted by the "misc_convdc" routine to a string at DECOUT (see Page Seven "Kernel" Values).

WRTCMD COMTAB-2

This location contains the SIO write command. A 'W' here indicates write with verify, while a 'P' indicates write without verify.

COMTAB COMTAB+0

This is a 6502 jump instruction followed by the address of the DOS entry routine. A jump here enters DOS.

ZCRNAME COMTAB+3

This is a 6502 jump instruction followed by the address of the file name crunch routine. This location is used to interpret the command line. A jump here will pull the next command from LBUF, translate the drive or device identifier if one is given (i.e., A: to D1:), add the default drive identifier if none is given, and place the result at COMFNAM. Each call will advance BUFOFF to point to the next entry on the command line, so that each call to the crunch routine will get the next entry on the line. If there are no entries remaining, the 6502 zero flag will be SET on return. Since the 6502 has no indirect jsr, it is necessary to use a few lines of code to access this routine. An example will follow this list.

BUFOFF COMTAB+10

The offset into LBUF where the next parameter to be read is located. This can be manipulated to reread the command line.

DATER COMTAB+13

The date in DD/MM/YY format (3 bytes). Updated by VGETTD. Updated continuously while the Time/Date line is on with SpartaDOS X.

TIMER COMTAB+16

The time in HH/MM/SS format (3 bytes). Updated by VGETTD. Updated continuously while the Time/Date line is on.

_800FLG COMTAB+27

SpartaDOS X only. \$FF if the computer is an Atari 800. Zero otherwise.

NBANKS COMTAB+29

SpartaDOS X only. The number of expansion memory banks free. This is the same number shown with the MEM command.

BANKFLG COMTAB+30

SpartaDOS X only. \$FF if USEing BANKED. Zero otherwise.

OSRMFLG COMTAB+31

SpartaDOS X only. \$FF if USEing OSRAM. Zero otherwise.

Note: USE NONE is indicated by both BANKFLG and OSRMFLG being zero.

COMFNAM COMTAB+33

This is the destination buffer for the ZCRNAME routine. It will ALWAYS begin with a Dd: since the default drive is added if none is given. If you are looking for switches or other options, start looking at COMFNAM+3. This buffer is 28 bytes long.

LBUF COMTAB+63

This is the input buffer for the command processor. The entire command line is stored here. LBUF is 64 bytes long.

COPYBUF COMTAB+127

This is the main buffer used by the SpartaDOS X "kernel".

An Example The following assembly language program demonstrates one way to read the SpartaDOS command line. It simply echoes the command line with the drive specifications added or translated as necessary. It resets BUFOFF to 0 so that the name of the command is printed, too.

```
    ; CIO and IOCB equates
    ciov .equ $E456
    iccom .equ $0342
    icbal .equ $0344
    icbah .equ $0345
    icbll .equ $0348
    icblh .equ $0349
    write .equ $09
    ; SpartaDOS equates
    comtab .equ 10
    zcrname .equ 3
    bufoff .equ 10
    comfnam .equ 33
    ; The program.
    *=$4000                ; or wherever.
```

Programming With SpartaDOS X

```
init                                ; patches our crunch routine to
    ldy #zcrname+2                  ; be the same as the COMTAB one.
    ldx #2

loop1
    lda (comtab),y
    sta crunch,x
    dey
    dex
    bpl loop1
; zero BUFOFF
    lda #0
    ldy #bufoff
    sta (comtab),y

mainloop
    jsr crunch                      ; get next command line entry.
    beq exit                        ; quit if there are no more.
; Set up for CIO print of data at COMFNAM
    ldx #0                          ; IOCB #0 (E:)
    lda #63                         ; set buffer length for max
    sta icbll,x
    lda #0
    sta icblh,x
    lda comtab                       ; store COMTAB+33 at icba
    clc
    adc #comfnam
    sta icbal,x
    lda comtab+1
    adc #0
    sta icbah,x
    lda #write                       ; 'print string' command
    sta iccom,x
    jsr ciov                         ; print it.
    jmp mainloop

exit
    rts

crunch
    jmp $FFFF                        ; will be changed by INIT routine
    *= $02E0
    .word init                       ;run vector
```


Vectors Under the OS ROM

The following vectors are only available on XL/XE computers. They reside under the Operating System ROM and will be invalidated by any program using this space, such as Turbo BASIC XL. It would be a good idea to check `_800FLG` to be sure it is an XL/XE computer and to check each vector before access to be sure that it is still there.

Since these vectors are under the OS ROM, it is necessary to enable the RAM instead of the ROM in this memory area. One possible method follows:

```
lda $D301           ; PIA, responsible for bank
pha                ; selecting. Store status.
and #$FE          ; RESET bit 0.
sta $D301         ; enable RAM under OS ROM.
jsr VGETTD        ; call the vector.
pla
sta $D301         ; restore PIA to previous state.
```

These functions each contain a jump (JMP) followed by the address of the function. It is a good idea to always check for this JMP before assuming that the vector is still there.

VGETTD **\$FFC0**

Returns the current time and date to COMTAB locations TIMER and DATER.

VSETTD **\$FFC3**

Sets the current time and date to the values held in COMTAB locations TIMER and DATER.

VTDON **\$FFC6**

Turns the time/date display on and off. If the 6502 Y register contains a 1 on entry, the time/date line will be turned on. If it contains a 0 on entry, the time/date line will be turned off. The 6502 carry flag will be set on return if the operation failed (TD.COM had not been run to install this routine).

VFMTTD \$FFC9

Returns a formatted time/date line to a specified buffer. On entry, the 6502 x register should hold the high byte of the buffer address. The Y register should hold the low byte of the buffer address. The 6502 carry flag will be set on return if the operation failed (TD.COM had not been run to install this routine).

VXCOMLI \$FFD2

Will cause the line contained in COMTAB buffer LBUF to be executed. BUFOFF should be 0 on entry.

VKEYON \$FFDB

Turns the key buffer on and off. If the 6502 Y register contains a 1 on entry, the key buffer line will be turned on. If it contains a 0 on entry, the key buffer line will be turned off.

Page Seven “Kernel” Values

Several page seven locations allow “kernel” operation to be accessed. While it is beyond the scope of this manual to document all of these locations, a few may prove to be of interest.

Name	Address	Function
sparta_flag	\$700	'S' if SpartaDOS
sparta_version	\$701	version in hex; \$32 = 3.2, \$40 = 4.0, etc.
kernel	\$703	jump (JMP) to “kernel” entry
misc	\$709	jump (JMP) to “misc” entry
sio_index	\$70F	“swap” table (9 bytes)
device	\$761	“kernel” device number
name	\$762	filename and ext (11 bytes)
date	\$77B	see below (3 bytes)
time	\$77E	see below (3 bytes)

dateset	\$781	see below
path	\$7A0	path only string (64 bytes)

The “kernel” routine is called by doing a subroutine jump (JSR) to address \$703 with the desired command in the 6502 Y register and the desired device number in *device*. For example, with a \$10 in *device*, a value of 100 in Y will cause the current time and date to be placed in the variables *time* and *date*. A 101 will cause the current time to be set to the values contained in the variables *time* and *date*.

kd_gettd	100	get current time and date
kd_settd	101	set time and date

The following is a list of valid “misc” vector commands. These should be loaded into the A register before executing a JSR \$709. The Y register is used as an index into COPYBUF for those operations using COPYBUF.

misc_initz	0	initialize misc driver
misc_getfina	1	convert path at COPYBUF to <i>device</i> , <i>path</i> , and <i>name</i>
misc_getpath	2	convert path at COPYBUF to <i>device</i> and <i>path</i>
misc_convdc	4	convert three byte number at DIVEND to a text string at DECOUT

The low nibble of the *device* number is the unit number of the device, such as 2 for D2:. The high nibble is one of the following:

0	SIO block device (SPARTA.SYS, ATARIDOS.SYS, etc.)
1	Clock driver (CLOCK.SYS, JIFFY.SYS)
2	ROM cartridge driver
3	Console driver
4	Printer driver
5	RS232 driver (COM.SYS)

6	reserved
7	reserved

Whenever a file is opened the time and date for that file will be placed in *time* and *date*. When a file is opened for write only and *dateset* equals 0, the current time and date will be read into *time* and *date* and assigned to the file. If *dateset* is -1 (\$FF), the file will get the time and date that are in the variables when the open is executed. *Dateset*, unlike the old COMTAB location TDOVER from previous versions of SpartaDOS, will automatically clear after use. This is how a copy of a file can retain the time and date of the original. This is also how a program like ARC assigns stored time/date information to a new file.

Technical Information

SpartaDOS Disk Format

There are four distinct types of sectors on a SpartaDOS format disk. These are boot, bit map, sector map, and data sectors. Data sectors may contain either directory data or file data. The following is a detailed discussion of each type of sector.

Boot Sectors As with most other DOS types for the 8-bit Atari computer, the first three sectors on the disk are boot sectors. These contain a program to load the file designated into the system when booted and other information needed to be able to store and retrieve data to and from the disk. The boot sectors are always single density, regardless of the density of the rest of the disk.

Sector 1 from offset \$30 to offset \$7F and all of sectors 2 and 3 are the boot code that loads a file under SpartaDOS 2.x and 3.x if specified (with the BOOT command). This code is not used with SpartaDOS X. The first part of sector 1 is a data table containing the values listed below as offsets into the sector. A *disk* can be a floppy disk, a RAMDISK, or a hard drive partition unless otherwise specified. All two or three byte numbers are stored in standard low byte/high byte format.

These are the sector 1 values, given as offsets into the sector:

- 9 The sector number of the first sector map of the MAIN directory.
2 bytes.
- 11 The total number of sectors on the disk (2 bytes).

- 13 The number of free sectors on the disk (2 bytes).
- 15 The number of bit map sectors on the disk.
- 16 The sector number of the first bit map sector (2 bytes).
- 18 The sector number to begin the file data sector allocation search. This is the first sector checked when an unallocated sector is needed. This serves two purposes; it relieves the necessity of searching the bit map from the beginning every time a file is to be allocated, and it allows sectors to be reserved after the main directory for directory expansion (2 bytes).
- 20 The sector number to begin the directory data sector allocation search. This is the first sector checked when a directory is to be expanded or added. Keeping this separate from the search number above will help keep directory sectors close together to speed searches (2 bytes).
- 22 The disk volume name. SpartaDOS uses this as part of the disk change identification procedure (8 bytes).
- 30 The number of tracks on the disk. If the drive is double-sided bit 7 will be set. If it is not a floppy disk (a RAMDISK or hard drive partition, for example) this location will contain a 1.
- 31 The size of the sectors on this disk (other than the boot sectors). A 0 indicates 256 bytes per sector, while a 128 indicates 128 bytes per sector.
- 32 The **major** revision number of the disk format. SpartaDOS 1.1 disks will have a \$11 here. Disks formatted for SpartaDOS 2.x, 3.x, and SpartaDOS X will all have a \$20 here, since they all use identical disk formats.
- 38 Volume sequence number. This number is incremented by SpartaDOS every time a file is opened for write on the disk. This is used to identify the disk.

- 39 Volume random number. This is a random number created when the disk is formatted. It is used with volume name and volume sequence number to positively identify a disk, to determine whether or not the data in the disk buffers is still valid.
- 40 The sector number of the first sector map of the file to be loaded when the disk is booted. This is usually a .DOS file. It is set by XINIT.COM from the SpartaDOS Construction Set and the BOOT command.

Bit Maps

A bit map is used to determine the allocation status of each sector on the disk. Each bit in every byte in the bit map shows whether the corresponding sector is in use, so each byte represents the status of eight sectors. Bit 7 represents the first sector of each group and bit 0 represents the eighth sector of each group. The bytes are in sequential order. Byte 0 of the first bit map sector represents sectors 0 through 7 (although sector 0 does not exist), byte 1 represents 8 through 15, and so on. If the bit representing a sector is SET (1), the sector is not in use. If it is CLEAR (0), then the sector is allocated. If more than one bit map sector is needed, any additional bit maps will follow on consecutive sectors.

Sector Maps

Sector maps are lists of the sectors that make up a file. The first two entries are the sector numbers of the next and previous sector maps of the file. The rest of the sector is a list of the sector numbers of the data sectors of the file or directory. The following are listed as offsets into the sector map:

- 0 The sector number of the next sector map of the file or directory. This will be 0 if this is the last sector map (2 bytes).
- 2 The sector number of the previous sector map of the file or directory. This will be 0 if this is the first sector map (2 bytes).

- 4 The sector numbers of the data sectors for the file in the proper order. If the sector number is 0, then that portion of the file has not been allocated. All sector numbers are two bytes long. See the Programming With SpartaDOS X chapter under the POINT command for a description of sparse files.

Directory Structure

The directory is a special file that contains information about a group of files and subdirectories. Each directory entry is 23 bytes in length and contains the file name, time/date, length, the number of the first sector map, and the entry status. The first entry is different from the others; it contains information about the directory itself. The following is a list of this information given as offsets into the first entry:

- 1 The sector number of the first sector map of the parent directory. A 0 indicated that this is the main (or root) directory of the disk (2 bytes).
- 3 The length of the directory (in bytes). This is the length of the directory file, not the number of entries (3 bytes).
- 6 The name of the directory padded with spaces (8 bytes).

When a directory is opened in unformatted or raw mode (see Programming With SpartaDOS X) the file is positioned to the second entry (that of the first file or subdirectory). To read the first entry you must POINT to the beginning of the file after opening it.

The rest of the directory entries are the same. They are 23 bytes long and provide the following information (given as offsets into the entry):

- 0 Status byte. The bits of this byte, if SET (1), represent the status of the directory entry as follows:

- B0 - Entry is protected.
- B1 - Entry is hidden.
- B2 - Entry is archived.
- B3 - Entry is in use.
- B4 - Entry is deleted.
- B5 - Entry is a subdirectory.
- B7 - Entry is open for write.

Notes: bits 1 and 2 are not supported by earlier versions of SpartaDOS. Bits 3 and 4 should always be opposites. Bit 5 should never be changed! A status byte of 0 indicates the end of the directory. Bits 6 is not used and should not be, since it may be cleared as other operations are performed.

- 1 The sector number of the first sector map of the file or subdirectory (2 bytes).
- 3 The length of the file in bytes (3 bytes).
- 6 The name of the file or subdirectory, padded with spaces if necessary (8 bytes).
- 14 The extension of the file or subdirectory, padded with spaces if necessary (3 bytes).
- 17 The date the file or directory was created in DD/MM/YY format (3 bytes).
- 20 The time the file or directory was created in HH/MM/SS 24 hour military format (3 bytes).

Exploring Disks

The best way to become familiar with the SpartaDOS disk format is to use a sector editor and a test floppy to explore. DiskRx, the SpartaDOS disk editor included in the SpartaDOS Toolkit, is an excellent sector editor tailored specifically for SpartaDOS disks. It will identify boot, bit map, sector map, directory, and data sectors. A good understanding of SpartaDOS disk structure and DiskRx can prove to be invaluable for recovering files from disks with bad sectors or damaged directories. Exploring disks can also be a lot of fun.

Configuring Your System

This chapter contains all the information needed to configure your system the way you want it. There are many drivers for various functions that can be installed into the system. Of course, if you install all of these, you may not have enough memory left to program with or load a particular application.

When SpartaDOS X boots, it has certain defaults — in fact, it contains a text “file” of configuration information. You may write your own file and override the default configuration “file”. The file you create is called “CONFIG.SYS” and should reside as a text file on drive 1 when you boot. It must be on a SpartaDOS format diskette in the “MAIN” directory.

CONFIG. SYS File The “CONFIG.SYS” file basically consists of commands. There are currently three commands. They are

```
USE OSRAM|BANKED|NONE
SET var=env_string
DEVICE driver
```

The USE command should be the first command in your “CONFIG.SYS” file as it instructs what secondary RAM area to use. OSRAM refers to RAM under the OS, BANKED refers the banks of RAM from \$4000-\$7FFF in your 130XE or memory expanded 800XL or 800 computer. NONE refers to low memory, above the normal low memory part of SpartaDOS and below program memory. USE NONE will probably be incompatible with many programs, since it uses up a great deal of main memory which may be used by applications.

The default ROM CONFIG.SYS pseudo-file is

SpartaDOS X Reference Manual

```
DEVICE SPARTA OSRAM
DEVICE SIO
DEVICE ATARIDOS
DEVICE INDUS
DEVICE CLOCK
DEVICE JIFFY
DEVICE RAMDISK
```

The default RAM usage is as follows:

OSRAM	Stock XL/XE computer
BANKED	Banked memory expanded computer (RAMBO XL, AXLON, or compatible upgrades)
NONE	800 with no expanded memory

Note that the size of the OSRAM memory area is 7K (\$E400-\$FFBF) and the BANKED memory area is 16K (\$4000-\$7FFF). If you have BANKED RAM, it is generally best to "USE BANKED" unless you have a stock 130XE and wish to use BASIC XE in EXTEND mode (or any other programs that require the extra 64K of RAM). If you do choose OSRAM, you may use the 4K of RAM from \$C000-\$CFFF as buffers for the SPARTA.SYS driver (explained later in this section).

The SET command is identical to the SET command in the command processor. You may set environment variables such as "CAR", "BASIC", or "BATCH" to default values.

The DEVICE command will load installable drivers such as SPARTA.SYS, CLOCK.SYS, etc. Each of these drivers is documented in this section. Note that order is important (e.g. SPARTA.SYS must load before ATARIDOS.SYS).

SPARTA.SYS Driver

- Purpose** This is the SpartaDOS format diskette driver. It must be installed — if it is not, your system will have no purpose (i.e. no way to read/write to disk).
- Syntax** `DEVICE SPARTA [OSRAM] [nbufs [,nfiles]]`
- Type** External — on device CAR:
- Remarks** This is the largest of all the drivers and contains 3 subprograms, these are 1) the SpartaDOS “kernel” functions, 2) the formatted directory output and other miscellaneous functions (the MISC vector), and 3) the default block I/O functions (the BLOCK_IO vector).
- The “OSRAM” parameter only applies if the system is set to “USE OSRAM”; it will be ignored otherwise. In this mode, the memory from \$C000-\$CFFF will be used for sector buffers, otherwise they will be allocated from main RAM. The default is to **not** use “OSRAM”.
- The “nbufs” parameter is the number of sector buffers to allocate, ranging from 2 to 16 — the default is 4. Each buffer will take up 256 bytes of RAM. This will be taken from \$C000-\$CFFF if you USE OSRAM add the OSRAM parameter, from the DOS bank if you USE BANKED, or from low memory (raising MEMLO) if you USE NONE or USE OSRAM but leave the OSRAM parameter off of DEVICE SPARTA (which you shouldn’t do, anyway).
- The “nfiles” parameter is the maximum number of disk files that may be open at one time, ranging from 3 to 16 — the default is 5. Each number here takes up 35 bytes of memory. If you USE BANKED this will be taken from the DOS bank. If you USE OSRAM and use the OSRAM parameter, this will be taken from \$C000-\$CFFF until that area is full and from low memory (raising MEMLO) after that. If you USE NONE (or USE OSRAM without using the OSRAM parameter for DEVICE SPARTA) this will be taken from low memory (RAISING MEMLO).

SIO.SYS Driver

Purpose	This is the high speed SIO and parallel I/O driver. It is a required driver as there is no default SIO driver.
Syntax	DEVICE SIO
Type	External — on device CAR:
Remarks	You must include this driver in your “CONFIG.SYS” file. It contains all the code to handle high speed SIO operations with the US Doubler 1050, Indus, Happy 1050, and XF551 drives. It also handles the standard speed SIO with all other drives and the standard parallel I/O (PIO) with devices such as the Multi I/O. DEVICE SPARTA must precede DEVICE SIO in CONFIG.SYS.

INDUS.SYS Driver

Purpose	This is the high speed SIO installer for Indus drives. It is required for high speed operation with Indus drives and Happy drives.
Syntax	DEVICE INDUS
Type	External — on device CAR:
Remarks	This driver takes up no memory, it simply programs all Indus drives with the appropriate high speed code. (The Indus drives have high speed code already in them except that it is buggy code and does not work.) Once the drive is programmed, it will stay programmed until power is shut off on the drive. Thus, you do not necessarily need to program the drives every time you boot your computer. Also, you must install the “SIO.SYS” driver before the “INDUS.SYS” driver. This driver is required for Happy Drives.

ATARIDOS.SYS Driver

- Purpose** This driver contains the code to recognize Atari DOS 2 format diskettes. This driver also supports the various derivatives of DOS 2 including MYDOS and DOS 2.5.
- Syntax** DEVICE ATARIDOS
- Type** External — on device CAR:
- Remarks** This driver requires that the “SPARTA.SYS” driver has been previously loaded (it is like an extension to the “SPARTA.SYS” driver). It supports all the derivatives of Atari DOS 2 including subdirectories of MYDOS. It supports the extended sectors of DOS 2.5 for read only. It does not support the ability to create (MKDIR) a directory, delete (RMDIR) a directory, or set working directory (CD) on MYDOS disks. ATARIDOS.SYS does not provide support for DOS 3, DOS XE, or OSS version 4 DOS.

RAMDISK.SYS Driver

- Purpose** This is the SpartaDOS X RAMDISK driver. You may select the drive and size of each RAMDISK you install.
- Syntax** DEVICE RAMDISK [drive] [,nbanks]
- Type** External — on device CAR:
- Remarks** The default parameters for this driver are: drive 9 and all available windowed RAM less 4 banks for BASIC XE use (default is all banks on the Atari 800). You may install multiple drives by using different drive numbers and selecting the appropriate sizes. If you attempt to select more banks of RAM than there are left, it will use all available RAM. (To see the number of banks of RAM your system has, use the MEM command.) Note that each bank is 16K bytes.

When the driver installs a RAMDISK, it will automatically build the directory structure on the RAMDISK. If you had previously installed the RAMDISK with the same size and drive number before performing a COLD start *without* loosing power to the RAM, it will not be reformatted (the contents will be preserved). Therefore, performing the following commands will not destroy the contents of your *RAMDISK*.

```
RAMDISK.SYS 8,4
COPY *.COM 8:
COLD (the system will reboot)
RAMDISK.SYS 8,4
DIR 8:
<all the COM files in drive 8 will be displayed>
```

A maximum of three RAMDISKS may be installed by RAMDISK.SYS. Any attempt to configure more will produce the error

```
RAMDISK not installed!
SIO device table full!
```

RAMDISK.SYS has no effect on Multi I/O RAMDISKS.

CLOCK.SYS Driver

- Purpose** This is the R-Time 8 driver for SpartaDOS X. Without this driver (or JIFFY.SYS) installed, the TIME/DATE commands give the SpartaDOS X revision time and date (which does not advance).
- Syntax** DEVICE CLOCK
- Type** External — on device CAR:
- Remarks** If there is already a clock device handler installed or if no R-Time 8 cartridge is plugged into the cartridge port, this handler will not load.

JIFFY.SYS Driver

- Purpose** This is the jiffy clock driver for SpartaDOS X. Use this clock driver if you don't have an R-Time 8 in your system. Without this driver (or CLOCK.SYS) installed, the TIME/DATE commands give the SpartaDOS X revision time and date.
- Syntax** DEVICE JIFFY
- Type** External — on device CAR:
- Remarks** If there is already a clock handler installed (such as CLOCK.SYS) this handler will not be installed.

XEP80.SYS Driver

Purpose This allows an 80 column display when use with the Atari XEP80 adaptor.

Syntax DEVICE XEP80

Type External — on device CAR:

Remarks The XEP80 must be plugged into the second joystick port and must have a monitor of its own. After installation, anything printed to the E: or CON: devices will be printed to the 80 column monitor through the XEP80. The regular 40 column output of the computer is unaffected, so that graphics output can be produced simultaneously.

It should be noted that many programs access screen memory directly and bypass the E: device. Almost all word processors are this way. These programs will not work through the XEP80, sending their output instead to the regular computer display. MENU.COM and FORMAT are good examples of this. Other programs use a combination of the two, such as the terminal program 850 Express! 3.0. The actual online part of Express! will work properly on the 80 column screen, but the menus will show up on the 40 column screen. For these reasons it is a good idea to keep the 40 column monitor attached and running if space permits. A good monochrome monitor is highly recommended for use with the XEP80.

Error Messages

The following is a list of error codes and messages that may occur while using SpartaDOS X. Some of the more common error codes are displayed in message form (as indicated by quotes) with most SpartaDOS X commands. Other programs may display error messages or just simply an error code (in either decimal or HEX (\$) form). Following each error code and message (if applicable) is a description of what probably caused the error. All error codes less than 128 (\$80) are application (BASIC, Action!, etc.) errors and are not produced by SpartaDOS X.

The descriptions here are meant to cover the most common error conditions. It is possible to get some of these error codes or messages under different circumstances but not likely.

128 \$80 "User break abort"

You pressed the <BREAK> key when the computer was waiting for input or printing to the screen. <BREAK> does not interrupt disk I/O in SpartaDOS X, but most programs will be terminated after disk I/O is completed if the <BREAK> key has been pressed.

129 \$81 "File already open"

You attempted to open a file for output which is already open. This can occur if you accidentally try to COPY a file on top of itself. For example,

```
COPY MYFILE
```

Since the default destination is "*", this error will occur. No prior versions of SpartaDOS made this type of check, so it was easy to inadvertently lose files using COPY.

This error can also occur when opening a file through the CIO if the IOCB had not been closed properly. This is a problem in some Atari programs (most notably the ACTION! cartridge). The command processor makes sure all IOCBs are closed when entered, so this error usually occurs within programs.

130 \$82 (Nonexistent device)

The device specifier you used does not exist. Valid device specifiers for the SpartaDOS X command processor are DSK:, CAR:, CLK:, PRN:, CON:, and COM:. Through the CIO, the valid devices are D:, E:, C:, S:, K:, R:, and P:. Of course devices may be added, but these are the standard devices.

131 \$83 (Channel or IOCB not open for input)

You attempted to read from a file that was open for write only (mode 8 or 9). This error would indicate a programming error.

132 \$84 “No device handler installed” (Bad CIO command)

You attempted to call the CIO with an invalid function code. Note that all function codes above 13 are considered XIO calls and therefore will not return this error. They return “No function in device handler” instead. This error would indicate a programming error. You may also get this error when attempting to access a “kernel” device with no handler installed such as COM:.

133 \$85 (Channel or IOCB not open)

You attempted to perform a read or write (or note/point XIO operation) through an IOCB or SpartaDOS X file channel that has not yet been opened. This indicates a programming error.

134 \$86 (Bad IOCB number)

You called the CIO with an invalid IOCB number in the X register. You must multiply the IOCB number you wish to use by 16. This indicates a programming error.

135 \$87 (Channel or IOCB not open for output)

You attempted to write to a file that was open for read only (mode 4). This indicates a programming error.

136 \$88 (End Of File)

This is not really an error but an indication of the end-of-file. This status may only be returned by a input function through the CIO. SpartaDOS X kernel calls return EOF status differently.

137 \$89 (Truncated record error)

This is not really an error but an indication that the record you attempted to read was longer than the buffer given to read the record into. This status may only be returned by an input function through the CIO. The SpartaDOS X kernel calls return this status differently.

138 \$8A "Drive not present"

You attempted to access a disk drive that was either non-existent, turned off, or disconnected. Also, your drives may have been SWAPped (see SWAP command). Check your SIO cables, power cords, and Multi I/O menu (if applicable).

139 \$8B "Drive NAK"

This error can occur under the following conditions: 1) Your disk drive door is open, 2) your Multi I/O is configured for a hard drive but none is online at that drive number, 3) you have a bad sector and your drive takes a long time to return any response. This can get the SIO out of sync and result in a "Drive NAK" error.

140 \$8C (Serial framing error)

This error indicates that your floppy drive and computer are not communicating properly. If this error consistently happens, then you probably need your drive or computer serviced. (It is possible that a serial framing error can occur if you have a bad sector, but it is unlikely.)

142 \$8E (SIO Serial bus overrun error)

This error indicates that your floppy drive and computer are not communicating properly. If this error consistently happens, then you probably need your drive or computer serviced. (It is possible that a serial bus overrun error can occur if you have a bad sector, but it is unlikely.)

143 \$8F (SIO Checksum error)

This error indicates that your floppy drive and computer are not communicating properly. If this error consistently happens, then you probably need your drive or computer serviced. (It is possible that a serial checksum error can occur if you have a bad sector, but it is unlikely.)

144 \$90 "Write protected or bad sector"

If you are reading from a disk, this error indicates that a sector is bad. If you are writing to a disk, you either have the disk write protected or the sector SpartaDOS is trying to write does not exist on the floppy (either because of a configuration problem or the sector has a bad header). Note that when you "Lock" a drive through the Multi I/O menu, you will get a "Drive NAK" error instead.

146 \$92 "No function in device handler"

You attempted to perform a command on a device that does not support that command. For example you cannot RENAME a file on the CAR: device or perform a directory listing of PRN:. On the CIO level, this indicates that the XIO function you attempted does not exist on the device specified.

148 \$94 "Unrecognized diskette format"

SpartaDOS could not recognize the DOS format of the disk you attempted to access. If the diskette is Atari DOS 2 format, then make sure that the ATARIDOS.SYS driver is installed in your system. (It is installed by default if you do not have a CONFIG.SYS on drive 1. If you have a CONFIG.SYS on drive 1, make sure that the line "DEVICE ATARIDOS" is included.)

150 \$96 “Directory not found”

You specified a directory path that does not exist. Recheck the pathname you specified. You may perform a directory command on each directory of your path to make sure that each directory exists.

151 \$97 “File exists”

You attempted to overwrite a file that is protected, replace a directory with a file, or replace a file with a directory.

152 \$98 “Not binary file”

You attempted to LOAD or run a file that is not a binary load file. There are several scenarios in which this error can occur.

1) The file does not start with a valid binary file header of \$FFFA or \$FFFF. (The file is a BASIC program, text file, database, etc.)

2) You attempted to run a relocatable SpartaDOS X command file with the X command. The X command only loads standard Atari binary load files.

3) The end of the file you attempted to load has been corrupted. This is generally caused by incompatible communications software when the file was either uploaded or downloaded from a bulletin board.

154 \$9A “Loader: Symbol not defined”

The SpartaDOS X loader could not load a program because it accessed a symbol that has not been defined. This indicates that you need to first load the appropriate driver for the command you are attempting. For example, the TD.COM command needs either the CLOCK.SYS or JIFFY.SYS drivers to be installed. These drivers define a symbol called I_GETTD which is referenced by TD to get the current time/date.

156 \$9C “Bad Parameter”

An invalid parameter has been given to a command. Refer to the appropriate command description in this reference manual for command syntax and usage.

158 \$9E “Out of memory”

You attempted to load or run a SpartaDOS X command that will not fit in memory. Make sure that there are no programs “held” in memory (see the LOAD command). If you are still out of memory, then reboot with fewer drivers and try again. The only case in which there is not enough memory available is when you attempt to ARChive files on an unmodified Atari 800 computer.

161 \$A1 (Too many channels open)

SpartaDOS X supports up to 16 open files, but each driver has its own limitation. The DSK: driver allows you to specify the maximum number of channels open to it (the default is 5 which should be enough for any application). The CAR: driver only has 1 channel which means that you may not copy files from CAR: with the COPY command. (COPY uses two channels on the source device and one on the destination. This is because COPY opens one channel to the directory and another to the file to be copied.) To overcome this limitation, you may TYPE the file on CAR: and redirect output to a file on disk (e.g. TYPE CAR:COMMAND.COM >>NEWCOM).

162 \$A2 “Disk full”

Your disk is full or your directory is full. SpartaDOS X directories handle up to 1423 files so it is probably a full disk. If you were copying files to the disk that became full, the file is removed from that disk.

163 \$A3 “Illegal wildcard in name”

You may not use wildcards when modifying or creating a file or creating a subdirectory. Wildcards are allowed when opening a file for input or in a directory path.

165 \$A5 “Bad filename”

The filename you entered has a bad character in it. The two most common places you will get this error are entering a bad character in a directory path or using a bad delimiter in the RENAME command.

166 \$A6 (Position range error)

You attempted to “point” beyond the end of file in a file that was open for input only. This error would indicate a programming error or too high a position when using the DUMP command.

167 \$A7 “Can’t delete directory”

The directory you tried to delete contains files or subdirectories. You must ERASE all files and delete all subdirectories including those that are hidden. Note: A file opened for write or update but not closed properly (usually due to a system reset or power loss while open) will leave a "phantom" entry in the directory. A subdirectory containing a "phantom" entry can not be deleted. You should use CLEANUP from the SpartaDOS Tool Kit to remove this entry to allow the directory to be deleted.

170 \$AA “File not found”

The file you tried to access does not exist. If you used the X command, remember that the PATH string is not used; you must include the full filename path if it is not in the current directory. This error will also occur if you attempt to rename or erase a protected file.

Command Summary — Alphabetical

This list is intended as a quick reference for command syntax and usage. For more details concerning command operation please refer to chapter 4.

ARC **command[option] [d:][path]arcfname[.ext] {filelist}**

Creates and maintains file archives.

Type - External — on device CAR:

ATR **[+A|H|P] [-A|H|P] [d:][path]fname[.ext]**

Sets/clears file attributes in the directory. Replaces the Protect and Unprotect functions from older SpartaDOS versions.

Alias - ATTRIB; Type - Internal

BASIC **[/N] [d:][path][fname] [parameters]**

Enters the *internal* BASIC in your XL or XE computer (1200XL does not have internal BASIC).

Type - External — uses CAR.COM on device CAR:

BOOT **[d:][path]fname[.ext]**

Tells a SpartaDOS formatted disk to load a specified file when the system is booted with this disk.

Type - Internal

CAR **[/N] [d:][path][fname] [parameters]**

Enters the cartridge plugged into the top of the SpartaDOS X cartridge. If a filename is specified, then that **binary** file is loaded and run with the cartridge enabled.

Type - External — on device CAR:

CHDIR [d:][path]

Changes the current (working) directory on the specified drive, or displays the current directory path if no path is given.

Alias - CD & CWD; Type - Internal

CHKDSK [d:]

Shows volume, free/total disk space, and sector size of the selected drive (or diskette).

Type - Internal

CHTD [+A|H|P|S] [-A|H|P|S] [d:][path]fname[.ext]

Changes the time/date stamp on all files matching the given filespec to the current time and date.

Type - External — on device CAR:

CHVOL [d:]volname

This command changes the volume name on the specified drive.

Type - External — on device CAR:

COLD [/CN]

Reboots the system (by doing a jump through \$E477).

Type - Internal

COMMAND (The Command Processor)

This program allows you to enter commands and run other programs. It is not entered as a command itself but is automatically invoked when you enter DOS.

Type - External — on device CAR:

COPY [d:][path][fname][.ext] [d:][path][fname][.ext][/A]

Copies one or more files to another drive and, optionally, gives the copy a different name if you specify it in the COPY command.

Type - Internal

DATE

Displays the current date and allows you to set the date.

Type - Internal

DIR [+A|H|P|S] [-A|H|P|S] [d:][path][fname][.ext] [/PC]

Displays a long formatted directory including byte size, date, and time.

Type - Internal

DIRS [+A|H|P|S] [-A|H|P|S] [d:][path][fname][.ext] [/PC]

Displays a short formatted directory (Atari DOS type).

Type - Internal

DUMP [d:][path]fname[.ext] [start] [len]

Displays a file in HEX and ATASCII form.

Type - External — on device CAR:

ERASE [d:][path]fname[.ext]

Deletes the file in the specified directory on the designated drive, or deletes the file from the default drive if no drive is specified. If no path is specified, the file is deleted from the current directory.

Alias - DEL & DELETE; Type - Internal

FIND [d:]fname[.ext]

Searches all directories on all drives for files matching the given filespec. If you enter a drive number, FIND will only look on that particular drive.

Type - External — on device CAR:

FORMAT

Initializes a disk in either SpartaDOS or Atari DOS 2 format. You may select density, sector skew, tracks, and volume name before formatting. It supports most known hardware configurations for your computer.

Type - Internal

KEY ON|OFF

Installs a 32 character keyboard buffer and also links an “internal” KEY command into your system (for turning the buffer on and off).

Type - External — on device CAR:

LOAD [d:][path][fname][.ext]

Loads a file (does not run). If no filename is used, all files previously loaded are removed from memory. This is useful for keeping commonly used commands resident in memory, thereby eliminating the need for these commands to load from disk.

Type - Internal

MEM

Displays the current low memory limits of your system and the number of available banks of windowed RAM.

Type - Internal

MENU

Allows you to select files and then perform COPY, ERASE, RENAME, etc. commands on all selected files. It is similar to other SpartaDOS menu programs, but provides many new features.

Type - External — on device CAR:

MKDIR [d:]path

Creates a subdirectory.

Alias - MD & CREDIR; Type - Internal

PATH [path_string]

Causes specified directories to be searched for commands before searching the current directory.

Type - Internal

PAUSE

Suspends system processing and displays the message “Press <RETURN> to continue”.

Type - Internal

PEEK [\$]location

Examines a memory location or performs a HEX conversion.

Type - Internal

POKE [\$]location [\$]value

Changes the contents of a memory location.

Type - Internal

PROMPT [prompt_string]

Change the system prompt.

Type - Internal

RENAME [d:][path]fname[.ext] fname[.ext]

Changes the name of one or more files.

Alias - REN; Type - Internal

RMDIR [d:]path

Deletes an empty subdirectory from the specified drive.

Alias - RD & DELDIR; Type - Internal

RS232

Loads the RS232 handler from a P:R: Connection or the Atari 850 interface.

Type - External — on device CAR:

SAVE [d:][path]fname[.ext] [\$]address [\$]address

Saves binary data from memory to disk.

Type - Internal

SpartaDOS X Reference Manual

SET [var[=env_string]]

To display the values of all environment variables and, optionally, sets an environment variable to a specified value.

Type - Internal

SWAP [d,d]

To swap (Re-map) your drive configuration or to display the current drive map list.

Type - Internal

TD ON|OFF

Turns on and off a time/date display line on top of your screen.

Type - External — on device CAR:

TIME

Displays the current time and allows you to set the time.

Type - Internal

TYPE [+A|H|P|S] [-A|H|P|S] [d:][path]fname[.ext] [/P]

Displays the contents of a specified file.

Type - Internal

UNERASE [d:][path]fname[.ext]

Restores files previously erased (if possible).

Type - External — on device CAR:

VERIFY ON|OFF

Turns write verify on or off.

Type - Internal

X [d:][path]fname[.ext] [parameters]

Executes a program which requires that no cartridges are installed (such as DISKRX, EXPRESS, most binary files, etc.)

Type - External — on device CAR:

Command Summary — By Function

This list is intended as a quick reference for command syntax and usage. For more details concerning command operation please refer to chapter 4 and 5.

Batch Files

-fname[.ext] [parm1 parm2 ... parmn]

Executes the specified batch file, optionally passing parameters. The extension is assumed to be .BAT if none is specified.

PAUSE

Suspends system processing and displays the message “Press <RETURN> to continue”.

Type - Internal

Directory Commands

DIR [+A|H|P|S] [-A|H|P|S] [d:][path][fname][.ext] [/PC]

Displays a long formatted directory including byte size, date, and time.

Type - Internal

SpartaDOS X Reference Manual

DIRS [+A|H|P|S] [-A|H|P|S] [d:][path][fname][.ext] [/PC]

Displays a formatted directory.

Type - Internal

CHDIR [d:][path]

Changes the current (working) directory on the specified drive, or displays the current directory path if no path is given.

Alias - CD & CWD; Type - Internal

MKDIR [d:]path

Creates a subdirectory.

Alias - MD & CREDIR; Type - Internal

RMDIR [d:]path

Deletes an empty subdirectory from the specified drive.

Alias - RD & DELDIR; Type - Internal

MENU

Allows you to select files and then perform COPY, ERASE, RENAME, etc. commands on all selected files. It is similar to other SpartaDOS menu programs, but provides many new features.

Type - External — on device CAR:

Disk Maintenance Commands

BOOT [d:][path]fname[.ext]

Tells a SpartaDOS formatted disk to load a specified file when the system is booted with this disk.

Type - Internal

CHKDSK [d:]

Shows volume, free/total disk space, and sector size of the selected drive (or diskette).

Type - Internal

CHVOL [d:]volname

This command changes the volume name on the specified drive.

Type - External — on device CAR:

FORMAT

Initializes a disk in either SpartaDOS or Atari DOS 2 format. You may select density, sector skew, tracks, and volume name before formatting. It supports most known hardware configurations for your computer.

Type - Internal

VERIFY ON|OFF

Turns write verify on or off.

Type - Internal

File Maintenance Commands

ATR [+A|H|P] [-A|H|P] [d:][path]fname[.ext]

Sets/clears file attributes in the directory. Replaces the Protect and Unprotect functions from older SpartaDOS versions.

Alias - ATTRIB; Type - Internal

COPY [d:][path][fname][.ext] [d:][path][fname][.ext][/A]

Copies one or more files to another drive and, optionally, gives the copy a different name if you specify it in the COPY command.

Type - Internal

ERASE [d:][path]fname[.ext]

Deletes the file in the specified directory on the designated drive, or deletes the file from the default drive if no drive is specified. If no path is specified, the file is deleted from the current directory.

Alias - DEL & DELETE; Type - Internal

RENAME [d:][path]fname[.ext] fname[.ext]

Changes the name of one or more files.

Alias - REN; Type - Internal

UNERASE [d:][path]fname[.ext]

Restores files previously erased (if possible).

Type - External — on device CAR:

MENU

Allows you to select files and then perform COPY, ERASE, RENAME, etc. commands on all selected files. It is similar to other SpartaDOS menu programs, but provides many new features.

Type - External — on device CAR:

Running Programs

BASIC [/N] [d:][path][fname] [parameters]

Enters the *internal* BASIC in your XL or XE computer (1200XL does not have internal BASIC).

Type - External — uses CAR.COM on device CAR:

CAR [/N] [d:][path][fname] [parameters]

Enters the cartridge plugged into the top of the SpartaDOS X cartridge. If a filename is specified, then that **binary** file is loaded and run with the cartridge enabled.

Type - External — on device CAR:

X [d:][path]fname[.ext] [parameters]

Executes a program which requires that no cartridges are installed (such as DISKRX, EXPRESS, most binary files, etc.)

Type - External — on device CAR:

Command Processor Options

COLD [/CN]

Reboots the system (by doing a jump through \$E477).

Type - Internal

COMMAND (The Command Processor)

This program allows you to enter commands and run other programs. It is not entered as a command itself but is automatically invoked when you enter DOS.

Type - External — on device CAR:

KEY ON|OFF

Installs a 32 character keyboard buffer and also links an “internal” KEY command into your system (for turning the buffer on and off).

Type - External — on device CAR:

PATH [path_string]

Causes specified directories to be searched for commands before searching the current directory.

Type - Internal

PROMPT [prompt_string]

Change the system prompt.

Type - Internal

RS232

Loads the RS232 handler from a P:R: Connection or the Atari 850 interface.

Type - External — on device CAR:

SET [var[=env_string]]

To display the values of all environment variables and, optionally, sets an environment variable to a specified value.

Type - Internal

SWAP [d,d]

To swap (Re-map) your drive configuration or to display the current drive map list.

Type - Internal

Time/Date Support

CHTD [+A|H|P|S] [-A|H|P|S] [d:][path]fname[.ext]

Changes the time/date stamp on all files matching the given filespec to the current time and date.

Type - External — on device CAR:

DATE

Displays the current date and allows you to set the date.

Type - Internal

TD ON|OFF

Turns on and off a time/date display line on top of your screen.

Type - External — on device CAR:

TIME

Displays the current time and allows you to set the time.

Type - Internal

Utilities and Programming Aids

ARC **command[option] [d:][path]arcfname[.ext] {filelist}**

Creates and maintains file archives.

Type - External — on device CAR:

DUMP **[d:][path]fname[.ext] [start] [len]**

Displays a file in HEX and ATASCII form.

Type - External — on device CAR:

FIND **[d:]fname[.ext]**

Searches all directories on all drives for files matching the given filespec. If you enter a drive number, FIND will only look on that particular drive.

Type - External — on device CAR:

LOAD **[d:][path][fname][.ext]**

Loads a file (does not run). If no filename is used, all files previously loaded are removed from memory. This is useful for keeping commonly used commands resident in memory, thereby eliminating the need for these commands to load from disk.

Type - Internal

MEM

Displays the current low memory limits of your system and the number of available banks of windowed RAM.

Type - Internal

PEEK **[\$]location**

Examines a memory location or performs a HEX conversion.

Type - Internal

POKE **[\$]location [\$]value**

Changes the contents of a memory location.

Type - Internal

SpartaDOS X Reference Manual

SAVE [d:][path]fname[.ext] [\$]address [\$]address

Saves binary data from memory to disk.

Type - Internal

TYPE [+A|H|P|S] [-A|H|P|S] [d:][path]fname[.ext] [/P]

Displays the contents of a specified file.

Type - Internal

Miscellaneous Notes

Using Turbo BASIC XL with SpartaDOS X

Turbo BASIC XL, a public domain language from Europe, has achieved great popularity in this country over the past few years. Since Turbo BASIC XL uses the RAM under the operating system ROM in XL and XE computers, it was not compatible with versions 2.x and 3.x of SpartaDOS. However, with the proper combination of hardware and configuration, however, Turbo BASIC XL will work well with SpartaDOS X.

Hardware Configuration It is necessary to have an XL or XE computer with more than 64K to use Turbo BASIC XL with SpartaDOS X. The Atari 800 has no RAM under the OS, making the running of Turbo BASIC XL on this computer impossible with any DOS. Unmodified XEGS, 1200XL, and 800XL computers are also unsuitable, since SpartaDOS X uses the RAM under the OS (USE OSRAM), conflicting with Turbo BASIC XL, or a large chunk of low memory (USE NONE), raising MEMLO too high to be used with Turbo BASIC XL.

System Configuration It is necessary to use a bank of extended memory (USE BANKED) to run Turbo BASIC XL. If you have more than 128K of RAM in your computer, this will occur by default. If you have just 128K, however, it is *required* that you boot with a custom CONFIG.SYS file on a SpartaDOS format disk in D1: to be able to use Turbo BASIC XL.

The first line of any CONFIG.SYS used while using Turbo BASIC XL *must* be

SpartaDOS X Reference Manual

```
USE BANKED
```

The rest of the CONFIG.SYS file is up to you. Do not forget to include DEVICE SPARTA and DEVICE SIO in this file. The following is an example of a CONFIG.SYS that will work with any XL/XE computer with more than 64K and Turbo BASIC XL:

```
USE BANKED
DEVICE SPARTA
DEVICE SIO
DEVICE INDUS
DEVICE CLOCK
DEVICE JIFFY
DEVICE RAMDISK
```

You can create the CONFIG.SYS file with a word processor or by typing

```
COPY CON: CONFIG.SYS
```

from the command line, typing in the above lines (making sure you press the RETURN key after each line), and then pressing the 3 key while holding down the CONTROL key. This is just an example. You may change it as you see fit, as long as the first line is USE BANKED.

Finally, don't forget to use X.COM when loading the Turbo BASIC XL main program, compiler, or runtime file. Use

```
X TURBO
X COMPILER
X RUNTIME
```

Using AUTORUN.SYS files

SpartaDOS X will not automatically load and run a file named AUTORUN.SYS when booted. With batch files and the relocatable nature of the SpartaDOS X command processor, however, the need for having an AUTORUN.SYS file is eliminated. There are three major types of AUTORUN.SYS you are likely to encounter. Following are descriptions of these and the best way to handle them.

Applications Many programs are named AUTORUN.SYS simply to have them load and run when the computer is booted. These files will usually be fairly long and will take control of the computer when run. To use these, simply rename them to a relevant name and type that name from the command processor. It may be necessary to use the X command to have the program perform correctly. It is not necessary to rename the program, but it is much more convenient to have the name of the file reflect the file's function and to be able to store several of these formerly AUTORUN.SYS files on one disk.

Handlers Many AUTORUN.SYS files install device handlers into the CIO's handler table. Among these are RS232 and other modem handlers and custom handlers, such as the G: device from ANALOG Computing (issue #35). These are usually short files and return control to the command processor or the language cartridge shortly after loading. These, too, can be renamed to some other name (such as RS232.COM or G.COM) and run from the command processor.

BASIC Program Loaders The third common type of AUTORUN.SYS file is a machine language program that loads and runs a BASIC program from disk. These are usually found on magazine disks. To use one of these, simply rename it something like MENU.COM and type

```
BASIC /N MENU
```

from the command line. Renaming the file is not required.

Using Batch Files Any of these programs or group of these programs can be made to run automatically by using batch files. Simply create a text file containing a list of the programs you wish to run and name it AUTOEXEC.BAT. When the computer is booted with this disk in D1:, the commands in the list will be automatically executed. For more information on batch files, please refer to chapter 8.

Using BASIC XE Extensions

The disk-based extensions for BASIC XE provide many useful tools for the programmer. They can also present a few problems for users of SpartaDOS X. Fortunately, these problems can be easily avoided.

Loading the Extensions The extension file "BASICXE.OSS" is loaded when the cartridge is entered for the first time. These extensions are loaded into the RAM area "under" the BASIC XE cartridge. However, if the time/date line is enabled (TD ON), it is very likely that problems will occur during the loading of the extension file. The solution to this is to either install the time/date line with a TD OFF command, leaving the actual display off, or to not install it at all.

Other Conflicts Once loaded, the extensions will still be there, whether you use internal BASIC or the X command to run programs. This can cause conflicts with the programs, and will almost certainly cause problems with attempting to use BASIC XE again. The best thing to do is to do a cold start with the COLD command to clear out the extensions if you plan to run other programs.

USING MAC/65 and DDT

MAC/65 works well with SpartaDOS X, but DDT, the debugger in the MAC/65 cartridge, will not operate properly with the key buffer active (KEY ON). The simple solution is to either do a KEY OFF before entering the cartridge or not installing the key buffer at all.

Problems with the Atari XF551 and Other Disk Drives

To understand how difficulties may arise when formatting disks it is first necessary to understand how the ICD Disk Formatter works.

Formatting Procedures The Formatter takes the configuration selected in the menu, executes a Set Drive Configuration SIO command, then executes a Format SIO command (which varies depending upon speed and density). If no errors are received the Formatter then writes a directory to the disk based upon the configuration data that you supplied.

Problems Problems can arise, however, if the drive accepts the configuration and format commands even though the drive is incapable of producing the format requested. As far as the Formatter knows, the drive successfully formatted in the requested configuration. The Formatter then writes a directory to the disk that is not correct, since no errors were received.

The XF551 The XF551 will not return errors if you give it an improper configuration. The following is a list of the valid configurations for the XF551. While you may not receive any errors formatting in other configurations, you may end up with a floppy disk with a bad directory. This could cause problems later as the disk fills up.

Density	Sides	Skew
Single	Single	Standard
Dual (enhanced)	Single	Standard
Double	Single or Double	Standard or High Speed

As you can see, doubled-sided and high speed formatting are *only* available when formatting in double density, regardless of the configuration entered into the Formatter menu! Also, note that single and dual (enhanced) density are only available single-sided format.

Other Drives This problem is not limited to XF551 drives by any means. The best solution is to make sure you know what configurations your drive is capable of accepting and only formatting in these.

Glossary

The following is a list of terms and their definitions as they apply throughout this manual and/or in many other computer publications.

- ADDRESS** A location in memory. An address may refer to a RAM or ROM storage location, a hardware register for an external device or processor, or a combination of these through bank selecting. For the Atari 8-bit, addresses range from 0 to 65535 (\$0000 to \$FFFF hex)
- APPEND** To add to. To append one file to another is to add the first onto the end of the second. This is often used when loading device handlers.
- ASCII** The American Standard Code for Information Interchange. This code uses seven bit data words (0 to 127 decimal, \$00 to \$7F hex) to define a standard character and control set. For example, the character S is represented by the number 83 (\$53 hex).
- ASSEMBLY LANGUAGE** Assembly language is a readable representation of the machine language in which the CPU is programmed. 6502 assembly language uses three letter mnemonics and operands to represent the numbers of the instructions and arguments. This assembly language source code is then translated by an assembler to the numbers that the 6502 can understand. Advanced assemblers allow the source code to include labels, macros (collections of often used instructions), and more.

SpartaDOS X Reference Manual

- ATASCII** The Atari 8-bit version of ASCII. ATASCII uses eight bits, providing codes from 0 to 255 (\$00 to \$FF hex) and displayable characters for almost every code. There are several differences between ATASCII and ASCII, the most notable being that ATASCII uses 155 (\$9B hex) as an EOL (end of line) marker, while ASCII uses a 13 (\$0D) for CR (carriage return) and 10 (\$0A) as an LF (line feed). Complete listings of ASCII and ATASCII values can be found in most programming reference books.
- BANK** A block of memory of specified size occupying a specific range of addresses. *Bank selecting* is the use of hardware registers to cause different banks of RAM, ROM, or hardware registers to occupy the same address space. This is necessary because the 6502, the CPU of the 8-bit, can only address 64K (65536 bytes) of memory. SpartaDOS X makes extensive use of bank selecting, allowing internal memory up to 1 megabyte (1024K or 1048576 bytes!) to be used effectively and easily.
- BATCH** Batch files are text files containing a list of commands to be executed consecutively.
- BAUD** A unit of measure of serial data transmission, named after the French inventor Jean-Maurice-Emile Baudot. This is the number of code elements per second. While this term is used interchangeably with bits per second (bps) in common usage, it is not actually the same.
- BINARY** The base 2 numbering system. Each digit of a binary number can be only a 0 or a 1. This is the numbering system used by all digital computers, since each digit may be represented by either the presence or the absence of voltage. Since binary numbers such as 01010011 can be tedious to work with, numbers are usually represented in decimal (83) or hex (\$53) format. A *binary file* is one that consists of numbers representing instructions and data directly readable by the computer. On the Atari, binary files also contain load addresses.

BIT	A single binary digit.
BOOT	The initialization of the computer, caused either by turning on the computer or by executing a cold start. The only difference in the two is that it is usually possible with SpartaDOS X to preserve RAMDISK contents after a cold start but not after losing system power.
BUFFER	An area of memory used as a temporary storage area for data. Buffers are commonly used for I/O (input/output) involving the keyboard, screen, disk drives, etc.
BYTE	A binary number consisting of 8 bits. Since the 6502 processes 8 bits of data at a time, most data in the Atari 8-bit is represented as one or more bytes. Each address points to one byte. These values are most easily referenced as a two digit hex number.
CENTRONICS	A standard parallel interface named for the company that first used it. Almost all parallel printers use "Centronics" type ports with a 'standard' 36 pin connector. The 8-bit Atari can be connected to a printer with a "Centronics" port only with an appropriate cable and an interface providing a parallel port, such as the Atari 850, the P: R: Connection, the Printer Connection, or the Multi I/O.
CIO	Central Input/Output. All communication with the screen, keyboard, and all peripheral devices may be handled through this part of the Atari operating system. The CIO is one of the things that sets the Atari 8-bit above other 8-bit computers.
COLD START	To cause the computer to initialize as if power were removed and reapplied without actually turning the computer off. This is faster than cycling power and will in most cases allow RAMDISK contents to be preserved with SpartaDOS.

SpartaDOS X Reference Manual

COMMAND	An instruction given to the computer from the user.
CP	Command processor. This portion of the DOS environment provides the interface between the user and the DOS. The CP prompts the user, interprets the commands given, and causes the specified operation to be executed.
CPU	Central Processing Unit. This is the main part of the computer, the part that reads and executes instructions. All programs must be translated into commands and data that the CPU can understand. The CPU for the Atari 8-bit computers is the 6502.
CRC	Cyclic Redundancy Check. This is a two byte number produced by performing a complex mathematical operation on a set of data. CRC is used in many applications, such as file transfer protocols and the ARC program.
CURRENT DIRECTORY	The directory assumed if none is specified. The default is the main, or root directory. The current directory may be changed to any subdirectory on a disk with the CHDIR command.
CURSOR	The mark on the screen that points to the place where the next action will take place.
CYLINDER	Used interchangeably with track, most often with hard drives, since these have multiple surfaces and heads.
DATA	Information used or processed by a program.
DEBUG	To isolate and correct errors in a program.

- DECIMAL** The base 10 numbering system. This is the numbering system used by human beings everywhere, consisting of numbers made of digits ranging from 0 to 9. While easier to understand, decimal numbers can be awkward to use for computer purposes.
- DEFAULT** The value or condition assumed if none is specified.
- DENSITY** Generally, this is the number of bytes in each sector of a disk. Single Density refers to 128 bytes per sector, while Double Density refers to 256 bytes per sector.
- DEVICE** An input and/or output interface to the computer, whether an actual physically external device, such as a printer, or a part of the computer simulating an external device, such as the screen editor. A program may access the D:, E:, S:, R:, P:, C:, K:, (the disk, screen editor, screen, serial port, printer, cassette drive, and keyboard, respectively) and other added devices through the CIO. SpartaDOS X, through the “kernel”, uses the DSK:, CAR:, CON:, COM:, and PRN: devices. These “kernel” devices may be accessed through the D: device from the CIO or independently through the command processor.
- DIRECTORY** The list of files and subdirectories disk or in a virtual disk (such as a RAMDISK or the CAR: device). If there are subdirectories on the disk, then it is the list in a given directory.
- DOS** Disk Operating System. The program which manages disk I/O to and from the computer. In practice, most DOS types also add functions and features to the operating system of the computer in areas not directly related to disk I/O.

SpartaDOS X Reference Manual

- DMA** Direct Memory Access. Any part of the computer system that can directly address system memory is considered to be a DMA device. The CPU is, of course, a DMA device. The only other DMA device on the Atari 8-bit system is ANTIC, the graphics coprocessor.
- DRIVER** A program that usually remains in the computer and handles a specific device or operation. SpartaDOS X includes, for example, a driver to allow it to read from and write to Atari DOS 2 format disks.
- FILE** A collection of information, usually stored as a named unit on a disk or virtual disk device.
- FIRMWARE** Software that is permanently encoded into ROM, preventing it from being changed or erased. SpartaDOS X is firmware, as are the computer's OS, the US Doubler, and any cartridge-based program.
- FORMAT** To initialize a disk so that it may be used to store and retrieve information. Physically the magnetic media on the disk is structured into tracks, which are divided into sectors. Directory information is usually written to a disk after it is physically initialized as part of the format operation. The SpartaDOS X command to format a disk is called **FORMAT**.
- HANDLER** An often memory resident program that handles a device. This usually patches into the CIO handler table to allow applications programs to access the device as it would any other. For example, most DOS types add a D: device handler. An R: device handler must be added for most serial I/O applications.
- HARD COPY** Information printed on paper.

- HARD DISK** A high capacity disk drive. Hard disks are usually sealed units, storing anywhere from 5 to 150 megabytes of information. Storing data on and retrieving data from a hard drive is usually many times faster than performing the same functions on a floppy disk, especially if the hard disk is connected to the computer through the PBI. Hard disks require the use of a hard disk controller, although some have the controller built into the drive assembly. A SCSI or SASI interface, such as the ICD Multi I/O, is required to connect the hard disk controller to the computer.
- HARDWARE** The computer, peripherals, and all related circuitry. Generally, anything that you can touch can be considered to be hardware.
- HEADER** Data at the beginning of a file that provides information about the type of file, how it should be run, and where it should be loaded.
- HEX** Short for HEXADECIMAL, the base 16 numbering system. It consists of numbers made of digits ranging from 0-9 and from A-F (representing 10-15). This is the easiest and clearest way to represent the eight and 16 bit binary numbers used in the Atari computer. To differentiate them from decimal numbers, hex numbers are usually preceded by a '\$' character.
- I/O** Input/Output. This refers to communication between the computer and the real world, including all devices and peripherals.
- ICD** The company that has written and designed power peripherals and software for the 8-bit Atari since 1984, including the US Doubler, P: R: Connection, Printer Connection, Multi I/O, R-Time 8, SpartaDOS Construction Set, SpartaDOS X, FA■ST Hard Drives, and more.

SpartaDOS X Reference Manual

- IOCB** Input/Output Control Block. A sixteen byte block of memory used to pass parameters to and from the CIO for I/O functions. There are 8 IOCBs starting at \$0340, numbered 0 through 7. IOCB 0 is normally used for the screen editor. A different IOCB must be used for each open device or file.
- K** Short for the metric prefix 'kilo'. A kilobyte is two to the tenth power bytes or 1024 bytes.
- KERNEL** The center of SpartaDOS X, responsible for I/O functions. The kernel provides I/O independent of the CIO and provides many new devices, such as CAR:, CON:, PRN:, and DSK:.
- KLUDGE** A program or part of a program or a hardware assembly that produces the desired results but does so in a complicated and/or unnecessary way.
- LANGUAGE** A program or development system that provides an easier or faster way to write a program. A program written in a programming language is then either translated into machine code (compiled) or is used as a set of commands for another program (interpreted) which then performs the desired tasks. Action!, BASIC, Pascal, and C are examples of popular languages.
- M** Short for the metric prefix 'mega'. A Megabyte is two to the twentieth power bytes or 1048576 bytes.
- MACHINE CODE** The program that the CPU reads and understands. All programs written in other languages must be translated into 6502 machine code (often called machine language) before they can be run. While the terms assembly language and machine language are often used interchangeably, they are not the same.

MEMLO	The address of the start of usable memory above DOS, memory resident handlers and other programs. The number is stored at memory locations 743 and 744 (\$2E7 and \$2E8). When using BANKED memory, SpartaDOS X provides the lowest MEMLO available, meaning that you have more programming space available.
MEMORY RESIDENT	A program that remains in memory after being run and continues to perform its task or tasks when necessary. DOS is memory resident, as are added device handlers. Most memory resident programs relocate to low memory and protect themselves from being overwritten by raising the low memory pointer.
MODEM	MODulator/DEModulator. A peripheral device that translates serial data from a computer into sounds that may be transmitted over telephone lines, allowing communication with similarly equipped computer systems at a distant location. Some MODEMS, such as the Atari 1030, XM301, and SX212 may be connected directly to an Atari 8-bit computer, but most require a serial interface such as the Atari 850 or the ICD P: R: Connection. <i>Hayes compatible</i> MODEMS accept a standard set of simple commands to perform a variety tasks, such as dialing, answering the phone, hanging up, etc.
MULTI I/O	An interface made by ICD that connects to the PBI and provides up to 1 megabyte of RAM, an RS232 serial port, a parallel printer port, and a SCSI/SASI compatible hard disk port. The MIO allows reassigning the logical drive units of physical floppy drives, hard drives, and RAMDISKS, so that the computer can be turned off and booted from any of these.
NIBBLE	Four bits, or one half of a byte. A nibble can be represented by a single digit hexadecimal number.
PATH	A list of subdirectory names describing the course from either the root directory or current subdirectory to a specific subdirectory.

SpartaDOS X Reference Manual

PARALLEL	The transfer, processing, or manipulation of all the bits in a byte simultaneously by using a separate line for each. This is usually faster than serial. Most printers are parallel devices.
PBI	Parallel Bus Interface, the large connector on the rear of the 800XL, allows the 8-bit Atari to communicate quickly with powerful external devices, such as the Multi I/O board. The PBI can be duplicated on the 130XE by a simple adapter connected to the cartridge and the ECI (Enhanced Cartridge Interface) ports.
PERIPHERAL DEVICE	Literally, a device on the periphery. In the computer world, a <i>peripheral device</i> , or just <i>peripheral</i> , is hardware added to the basic system configuration. Printers and MODEMs, for example, are peripheral devices.
PORT	A place of access to a system; e. g., the joystick port, the parallel bus interface port, the serial I/O port, printer port, cartridge port, etc.
PROGRAM	A set of instructions that cause a task to be performed by a computer. Programs must adhere to the order and conventions of the language in which they are written.
PROMPT	A signal to the user that some action may be required. The D1: or A: prompt with SpartaDOS X tells you that the computer is ready for input.
RAM	Random Access Memory. The storage area to which the computer may save and from which it may retrieve information. RAM will lose the stored information when power is removed.
RAMDISK	A specified area of RAM that simulates a disk drive. A handler or driver is used to make this memory appear to be a disk drive. Since they are actually RAM, RAMDISKS will lose their contents when power is removed.

- REAL TIME** Relating to real world time. A real time clock uses the actual time. Real time can also refer to things that occur at the same time or at realistic speeds.
- RELOCATABLE** A program that can be moved to different areas in memory and still operate properly. SpartaDOS X is relocatable. Most SpartaDOS drivers and handlers are self-relocating, usually meaning that they relocate themselves to low memory and move the low memory pointer MEMLO just above the code.
- ROM** Read Only Memory. ROM is like RAM except that it can not be changed and will remain the same even after a loss of power. SpartaDOS X, all of the programs in CAR:, and the Atari OS are in ROM.
- RS232** A standard serial communications interface documented by the Electronics Industries Association. Most MODEMs use an RS232 interface. The Atari 8-bit does not have an RS232 port, but one may be added with an Atari 850 interface or a P: R: Connection.
- SCSI** Small Computer System Interface. Most hard drives (with the notable exception of IBM PC specific hard drives) use either a SCSI or SASI (Shugart Associates System Interface) bus. Up to eight SCSI devices may be connected to a SCSI port on a computer. The Multi I/O from ICD provides a SCSI port for 800XL and 130XE computers.
- SECTOR** The standard block of storage used on disks. Sectors on the 8-bit Atari may contain either 128 or 256 bytes.
- SERIAL** The transfer of information on one signal line, one bit at a time. Most MODEMs and all devices connected to the serial I/O port (SIO) on the Atari are serial devices.

SpartaDOS X Reference Manual

SIO	Serial Input/Output. All communication with devices connected to the serial bus of the Atari computer are handled through this routine. Devices on the parallel bus which simulate SIO devices (such as MIO ramdisks and hard drive partitions) are also access through the SIO. The Atari OS provides an SIO routine. SpartaDOS, however, uses its own SIO code.
SOFTWARE	Programs, documentation, and data files that allow a computer to function.
SPARTA	A powerful, disciplined, war-loving city-state of ancient Greece.
SUBDIRECTORY	An additional directory on a disk allowing better organization through the grouping of associated files. Subdirectories are treated as entries in existing directories.
SYNTAX	The order and wording of commands or statements.
TRACK	A circular section of a disk surface. Each track is divided into sectors. A standard Atari 8-bit floppy disk is formatted into 40 concentric tracks each containing 18 sectors.
TRUNCATED	Shortened. Usually used when the provided amount of data exceeds the expected amount and what can not be accepted is discarded.
VARIABLE	A symbol that represents a quantity that is changeable or has no fixed value.
VIRTUAL DISK	Something that appears to the system and user to be a disk drive but is not. RAMDISKS, both internal and in the Multi I/O, and the CAR: device in SpartaDOS X are examples of virtual disks.
WARM START	A system reset that does not clear out all memory as a cold start does. Warm starts do reset several system pointers.
WILDCARDS	A symbol that is used as a substitute for one or more characters in a file or directory name to allow more than one file to be selected. '*' and '?' are the two valid wildcards with SpartaDOS and most others.

WORD	In the most common usage, a word is a sixteen bit or two byte number.
XIO	Extended Input/Output. A general I/O statement available in most Atari languages that allow CIO operations to be performed that are not supported by specific commands.

I n d e x

- Archive
 - See Commands, ARC
 - See also Commands, ATR
 - See also Attributes, file
- Assembly Language 6-1, 6-12, 6-14, 6-16,
..... 6-18
 - MAC/65 3-2, 4-38, 4-58
- Computer, Atari
 - 1200XL 2-2, 2-4, 4-9
 - 130XE 2-2, 2-11, 4-9, 4-40
 - 65XE 2-2, 2-11, 4-9
 - 800 2-2, 2-4 - 2-5
 - 800XL 2-2, 2-11, 4-9
 - XEGS 2-2, 2-11
- Attributes
 - See also Commands, ATR
 - See Directory, Attributes
 - File 6-3 - 6-4, 6-11, 4-7 -
..... 4-8, 4-19, 4-27 - 4-28,
..... 4-44, 4-46, 4-63
- Banks
 - See Random Access Memory (RAM)
- BASIC
 - See also Programming Languages
- Batch files 3-2, 4-1 - 4-2, 4-23,
..... 4-49, 4-51, 5-1 - 5-4
 - AUTOEXEC.BAT 5-2, D-2
 - Creating 2-13 - 2-13, D-2
 - Running BASIC Programs 5-2 - 5-4
 - Parameters 5-2 - 5-4
- Bit map 7-1 - 7-3, 7-5
- Buffers 8-2 - 8-3
- Central Input/Output (CIO) 3-5 - 3-7, 6-2, 6-12 -
..... 6-15, 6-18 - 6-19
- Command Processor (CP) 2-2 - 2-3
- Prompt 2-2 - 2-4, 4-2, 4-44,
..... 4-52 - 4-54, 5-1
- Commands
 - ARC 4-3 - 4-6
 - ATR 4-7 - 4-8, 4-19, 4-28
 - BASIC 4-9, 4-14 - 4-15, 4-66
 - BOOT 4-12, 4-34
 - CAR 4-9, 4-11, 4-14 - 4-15,
..... 4-38, 4-47, 4-59, 4-66
 - CHDIR 4-16, 4-56
 - CHKDSK 4-18 - 4-19
 - CHTD 4-19, 4-21, 4-47
 - CHVOL 4-19
 - COLD 4-10, 4-15, 4-20, 4-60
 - COMMAND 4-15, 4-21, 4-38 - 4-39
 - COPY 4-8, 4-12, 4-23 - 4-25,
..... 4-28, 4-41, 4-55
 - DATE 4-26
 - DIR 4-7, 4-18 - 4-19, 4-27 -
..... 4-28, 4-63
 - DIRS 4-27 - 4-28
 - DUMP 4-21, 4-29, 4-47
 - ERASE 4-8, 4-12, 4-28, 4-30,
..... 4-41, 4-55, 4-64
 - FIND 4-31, 4-66
 - FORMAT 4-12, 4-19, 4-32 - 4-34,
..... 4-36
 - KEY 4-37
 - LOAD 4-10, 4-15, 4-21 - 4-22,
..... 4-38 - 4-39, 4-58
 - MEM 4-39 - 4-40
 - MENU 4-23, 4-25, 4-30, 4-41,
..... 4-47, 4-66
 - MKDIR 4-46, 4-56
 - PATH 4-47 - 4-48
 - PAUSE 4-49
 - PEEK 4-50

SpartaDOS X Reference Manual

- POKE 4-51
- PROMPT 4-52, 4-54
- RENAME 4-55
- RMDIR 4-46, 4-56
- RS232 4-57
- SAVE 4-25, 4-38, 4-58
- SET 4-9 - 4-10, 4-15, 4-47 -
..... 4-48, 4-54, 4-59
- SWAP 4-60
- TD 4-61
- TIME 4-62
- TYPE 4-7 - 4-8, 4-24 - 4-25,
..... 4-28, 4-63
- UNERASE 4-55, 4-64
- VERIFY 4-65
- X 2-11 - 2-12, 4-15, 4-38 -
..... 4-39, 4-47, 4-66
- CONFIG.SYS 4-26, 4-40, 4-61 - 4-62,
..... 8-1, 8-4

- Device**
 - Cartridge (CAR:) 3-6, 4-2, 4-19, 4-21, 4-47 - 4-48
 - Console (CON:) 3-5 - 3-6, 4-24
 - Disk (D:, DSK:) 3-5 - 3-7, 4-24
 - Editor (E:) 3-5 - 3-7
 - Identifier 3-1, 3-5 - 3-7
 - Printer (P:, PRN:) 3-6 - 3-7, 4-4, 4-24 - 4-25
 - Serial (R:, COM:) 3-6, 4-57
 - Specifier 6-1

- Directory**
 - Current (Default) 3-
 - Disk 3-2 - 3-4, 3-6 - 3-8
 - Path 3-1, 3-4 - 3-5, 4-1 - 4-2,
..... 4-42 - 4-43, 4-47 - 4-48,
..... 4-52 - 4-53, 4-55 - 4-56,
..... 4-66

- Disk Operating System (DOS) 2-**
 - Drive, Default 3-2 - 3-4, 3-6 - 3

- Driver**
 - ATARIDOS.SYS 8-2, 8-6
 - CLOCK.SYS 4-26, 4-61 - 4-62,
..... 8-2, 8-8
 - INDUS.SYS 8-2, 8-5
 - JEFFY.SYS 4-26, 4-61 - 4-62, 8-2,
 - RAMDISK.SYS 8-7
 - SIO.SYS 8-2, 8-4 - 8-5, 8-7
 - SPARTA.SYS 8-2 - 8-4, 8-6
 - XEP80.SYS 8-9

- Environment Variables** .. 4-9 - 4-10, 4-14 - 4-15,
..... 4-59, 5-2, 5-4
 - BASIC 4-9 - 4-10
 - BATCH 5-2
 - CAR 4-14 - 4-15, 4-59
 - PATH 4-47 - 4-48, 5-4
 - PROMPT 4-54

- FlashBack!** 4-8, 4-28

- Format**
 - See also Commands, FORMAT
 - Directory 2-5, 6-2 - 6-4, 7-4
 - Disk 2-3, 2-14, 7-1 - 7-3, 7-5
 - Skew 4-33 - 4-34
 - Time and date 2-5, 4-26, 4-62, 7-5

- Happy Drives** 8-4 - 8-5

- I/O Redirection** 4-1 - 4-2, 4-4, 4-9, 4-66,
..... 5-1, 5-3

- Kernel** 6-2, 6-18, 6-21 - 6-22

- MEMLO 4-10, 4-15
- MODEM 4-6
- Multi I/O (MIO) 4-18, 4-60, 8-4

- Parameters** 2-6, 2-8, 5-1 - 5-2

- Path**
 - See Directory, Path

- Programming Languages**
 - Action! 3-2, 4-14, 6-1, 6-7

- BASIC 2-4 - 2-5, 2-9, 2-11,
 2-13 - 2-14, 3-2, 3-5 -
 3-7, 4-9 - 4-11, 4-15,
 4-20, 4-40, 4-66,
 5-3, 6-1, 6-3 - 6-4,
 6-7, 6-12, 6-14, 6-20
- PROTECT
 See Commands, ATR
 See also Directory, Attributes
- R-Time 8 2-2, 4-26
- Random Access Memory (RAM) 2-2, 4-2, 4-10,
 4-15, 4-20, 4-23, 4-25,
 4-32 - 4-33, 4-36, 4-39 -
 4-40
 Banks 2-2, 4-2, 4-5, 4-20,
 4-39 - 4-40, 6-17, 6-20
 8-1, 8-7
 RAMDISK 2-2, 4-39 - 4-40, 4-49
- Read Only Memory (ROM) 3-5 - 3-6
- Real Time clock
 See R-Time 8
- Sector 2-3, 2-5, 4-12 - 4-13,
 4-18, 4-27 - 4-29, 4-32 -
 4-35, 4-64, 7-1 - 7-5
 Editor 2-12, 4-29, 4-55, 7-5
 Map 7-1, 7-3 - 7-5
- Sector Maps
 See Sector
- Serial Input/Output (SIO) . 4-34, 4-51
- SpartaDOS Tool Kit
 DISKRX 2-12, 4-29, 4-55, 7-5
- Subdirectories 2-9 - 2-11, 2-14, 3-3 -
 3-4, 4-7 - 4-8, 4-12,
 4-16 - 4-17, 4-27 -
 4-28, 4-31, 6-4, 6-9,
 6-11
 See also Directory, Attributes
- Syntax 4-1, 4-3, 4-7, 4-28
- Time/Date 2-2, 2-5 - 2-7, 2-9, 2-14,
 4-4, 4-6, 4-10, 4-15,
 4-19, 4-26 - 4-28, 4-31
 See also Commands, CHTD
 See also Commands, TD
- UltraSpeed 4-33
- User interface 2-2
 See also Command Processor (CP)
- Wildcards 2-7 - 2-8, 2-10, 3-2, 4-1,
 4-24, 4-30 - 4-31
- XEP80 8-9
- XIO 6-1, 6-4 - 6-12


Error Message Summary

128	\$80	“User break abort”
129	\$81	“File already open”
130	\$82	(Nonexistent device)
131	\$83	(Channel or IOCB not open for input)
132	\$84	“No device handler installed” (Bad CIO command)
133	\$85	(Channel or IOCB not open)
134	\$86	(Bad IOCB number)
135	\$87	(Channel or IOCB not open for output)
136	\$88	(End Of File)
137	\$89	(Truncated record error)
138	\$8A	“Drive not present”
139	\$8B	“Drive NAK”
140	\$8C	(Serial framing error)
142	\$8E	(SIO Serial bus overrun error)
143	\$8F	(SIO Checksum error)
144	\$90	“Write protected or bad sector”
146	\$92	“No function in device handler”
148	\$94	“Unrecognized diskette format”
150	\$96	“Directory not found”
151	\$97	“File exists”
152	\$98	“Not binary file”
154	\$9A	“Loader: Symbol not defined”
156	\$9C	“Bad Parameter”
158	\$9E	“Out of memory”
161	\$A1	(Too many channels open)
162	\$A2	“Disk full”
163	\$A3	“Illegal wildcard in name”
165	\$A5	“Bad filename”
166	\$A6	(Position range error)
167	\$A7	“Can’t delete directory”
170	\$AA	“File not found”





SpartaDOS X


is the most progressive disk operating system ever designed for the Atari 8-Bit computer. Its magnified approach to power, support and generous DOS amenities are destined to expose limitless opportunities for all Atari 8-Bit computers, including the original Atari 800 when upgraded with an Axlon-compatible memory board.



This manual provides you with an in-depth look into the sophisticated world of SpartaDOS X, beginning with the basics of DOS to a technical analysis of the SpartaDOS X disk and file structure. The manual also includes a wealth of information on the command set, batch files, I/O redirection, search paths and more, so that you may take full advantage of SpartaDOS X's outstanding power and features. In little time you will be able to perform complex tasks with ease and configure your system for optimum use.



Not only does SpartaDOS X cater to the entire 8-Bit line, but also employs full use of extra memory with RAMdisks supported up to 1 MB and high speed operation when teamed with either Indus GT or Atari XF551 drives. SpartaDOS X's lightning-quick ARC utilities are credited for storing, compressing and extracting files with record setting speed and efficiency.



These qualities and many more rank SpartaDOS X as the disk operating system of the future... for today's Atari 8-Bit user.

