

DISK DETECTIVE

For the ATARI* 400/800 16K and one or more disk drives.

OVERVIEW

Disk Detective is a utility program that unlocks the mysteries of your ATARI disk system. Until now, you were limited by DOS commands to only a fraction of the power of your disk system's capabilities. In order to use this program properly, you should have a working knowledge of ATARI DOS and hexadecimal to decimal conversion.

Disk Detective consists of two unique program modules, DISKMAP and DETECTIVE. DISKMAP will let you see which sectors on your diskette contain data and which are free. DETECTIVE then lets you examine, modify, or move sectors as you desire.

LOADING DISK DETECTIVE

Assuring that your BASIC language ROM cartridge is inserted, boot the disk normally. When the READY prompt appears, type RUN"D: LOGO" and press RETURN.

When the MENU appears, select (1) for DETECTIVE or (2) for DISKMAP.

NOTE: Use DISK DETECTIVE with drive ONE only.

DETECTIVE

DETECTIVE gives you the power to examine, modify or move any sector recognized by ATARI DOS. After selecting this option from the main menu the following screen display will appear:

1. READ A SECTOR
2. DISPLAY BUFFER
3. WRITE A SECTOR
4. ZERO BUFFER
5. CHANGE BUFFER
6. EXIT PROGRAM

*Atari is a registered trademark of Atari Computer, Inc.

DATASOFT, INC.™

19519 Business Center Dr., Northridge, CA 91324

1. READ A SECTOR

This command reads a designated sector from the diskette and places its contents in a temporary storage buffer area.

There are 720 sectors available to the ATARI 810 disk drive. Unfortunately, a mis-communication has resulted in a numbering conflict between ATARI hardware and software. At this time, you will not be able to access sectors "0" or "720" until this problem is rectified. (With some DOS releases, sector 719 will not respond either).

Upon selecting this option you will be asked to input the sector number to read. You may read any sector between 1 and 719. Input the sector number desired, then press RETURN (cr). The sector will be read and your screen will return to the main menu.

2. DISPLAY BUFFER

When the sector has been read, select the DISPLAY BUFFER option. This option will PRINT the contents of the buffer area on your video screen.

The 128 bytes within the chosen sector are displayed in 16 lines containing 8 bytes per line. Each line displays the information in hexadecimal on the left side and in ASCII on the right side of your screen. The number in the far right margin indicates the beginning byte number for that line.

The hexadecimal and ASCII columns are numbered 0 to 7 at the bottom of the display. To find a specific byte location, add the column number to the number in the far right margin of that line.

3. WRITE A SECTOR

After modifying a sector you will want to write it back to the diskette. When you select this command you can write the contents of the buffer to the same sector it was read from, or any other available sector.

When the prompt appears, input an "N" for NO if you wish to return to the main menu. Inputting a "Y" for YES will write the current contents of the buffer to the sector it was read from. Inputting an asterisk "*" will allow you to write the data to other sectors on the diskette.

4. ZERO BUFFER

This command sets the flag byte to \$80 which appears to be a deleted sector with no data in its contents. You may wish to use this feature to prevent a file from being examined by others. Changing the flag byte will restore the sector whenever you decide to.

5. CHANGE BUFFER

Using this command alters the data on a given sector. You will first be asked to input the byte you wish to change (0-127). The location entered will be displayed in Hexadecimal and ASCII. You may then enter the change in either Hexidecimal or ASCII. To enter ASCII characters, type a single quote followed by the character (Example: 'M) and press RETURN (cr). Enter Hexidecimal code as it normally appears (00-FF), then press RETURN (cr). You may change as many bytes within a sector as you wish in this mode.

6. QUIT

Simply clears the screen and exits this program.

DISK MAP

DISK MAP prints a picture of the contents of your diskette on the screen for your examination. The display will provide 20 rows of 36 characters representing two tracks per row. Each track contains 18 sectors. Sectors containing data will be displayed with an asterisk (*). Open sectors will be displayed as dots (.) on your screen. To find the sector you wish to examine, count from the left of the row and add that number to the number displayed at the far right of the screen.

The DISK MAPPING process takes approximately 3 minutes. Upon completion, pressing RETURN (cr) ends this portion of the program. In order to keep this program within a 16K boundary, it was necessary to use many of the same areas of memory for DISK MAP as are used for DETECTIVE.

When you use this program, it is assumed that you intend to use a diskette other than DISK DETECTIVE for your examination and modification. Modifying your DISK DETECTIVE diskette can void your warranty. Please read the terms and conditions of sale for warranty information.

DOS NOTES

The Disk Operating System uses eight sectors for its VTOC (Volume Table Of Contents), beginning at sector number 359 and proceeding as files are added.

Catalogue information is arranged in the following manner:

BYTE DESCRIPTION

- 0 flag byte
 - 1 sector count (low)
 - 2 sector count (high)
 - 3 starting sector (low)
 - 4 starting sector (high)
-

- 5 file name
 - 6 " "
 - 7 " "
 - 8 " "
 - 9 " "
 - 10 " "
 - 11 " "
 - 12 file name
-

- 13 file extension
 - 14 " "
 - 15 " "
-

There may be up to 64 files on a diskette. The flag byte is interpreted as follows:

- \$00 — No File
- \$40 — In Use (closed)
- \$41 — In Use (open)
- \$60 — In Use (locked)
- \$80 — Deleted

Within the data sectors, bytes 125-127 maintain the data chain. In byte 125, bits 2-7 contain the file number while bits 0-1 contain the high value of the forward pointer. Byte 126 is the low value forward pointer, while byte 127 is the checksum with bit 7 indicating if less than 125 bytes of data are contained in the sector. The forward pointers will equal zero when the last sector is encountered.