BW-DOS 1.4

10-06-2023

User Manual

by BEWESOFT

Revised by
holgerjanz@abbuc.social
18-08-2023

Content

Introduction

BW-DOS is a disk operating system for 8-bit ATARI computers with at least 48kB RAM, and at least one disk drive. It is designed to be compatible with SpartaDOS, and to use as little memory as possible. In addition, several small bugs known from SpartaDOS are not in BW-DOS, especially problems with XF 551. BW-DOS may be used separately, or as a "junior version" for SpartaDOS users.

BW-DOS may be freely used and distributed under following conditions:

- Only the whole and unchanged master disk may be distributed,
- It may not be sold, excepting little costs in PD services,
- When used as a part of another software package, the whole package may be marked with a "(C)", sold etc.; in this case it's possible to exclude files from BW-DOS, but all distributed files still must be unchanged, and an information where the whole BW-DOS is available must be included.

BW-DOS uses disk format compatible with SpartaDOS. It supports SpartaDOS File System version 2.0 with an extension that supports more than 126 entries per directory (see chapter Disk and Directories). That means it is possible to use BW-DOS disks under SpartaDOS (X), and SpartaDOS (X) disks under BW-DOS in full (including booting). Most of CIO commands and other features are also compatible, so a lot of programs written for SpartaDOS will work under BW-DOS without any problems. Any disk with capacity up to 16 megabytes is possible, single file may be up to 8 megabytes long.

BW-DOS is a command driven system - it is more flexible than a common DUP menu. An advanced menu program is available too.

Up to 5 files may be open at the same time, and five disk drives are supported (1, 2, 3, 4 and 8). This is enough for most of programs.

BW-DOS does not use the RAM under OS-ROM (XL/XE machines only), so it is compatible with software which uses this memory space, such as popular Turbo Basic, and others. The MEMLO value is $1ECF (BW-DOS version 1.4); this value may only be changed by resident commands. Don't think that it is too high - BW-DOS is much more complicated than - for example - Atari DOS 2, and besides of this, when you set other DOSes to 5 files open at the same time, and 5 disk drives, you'll get MEMLO far above $2000. BW-DOS supports MEMLO up to $3000 e.g., for installing resident commands or drivers. If MEMLO is above $3000 some external commands will not work any longer because they usually start at $3000 so that these commands also woks on 16k machines.

Because of limited memory space, BW-DOS doesn't support some of the functions known from SpartaDOS as internal functions. They are supported by external and resident commands. Atari DOS 2 disks are only supported by the menu program.

If you have found a bug in BW-DOS, you can write to the author. Send only serious letters, please! (I'll not answer others.) The address is:

Jiří Bernášek (BEWESOFT) Na Hrebenkach 42 150 00 Praha 5 Czech Republic

If you have found some spelling or other errors in this manual, please forgive them... (I'm not very good on English).

Holger Janz (holgerjanz@abbuc.social): The original source of version 1.3 was provided by Jiří.

Command Syntax

In this manual there are examples, and syntax definitions for several commands. When it is between another text, it is written like: "example". Don't forget that you need to press the <RETURN> key after every command line. Following syntax is used in the examples and definitions:

<KEY> - This tells you that you need to press specified key. "..." (three dots) means "and so on".

UPPER CASE LETTERS, and other characters such as numbers etc. must be written exactly as shown in the manual.

lower case letters - It must be substituted by corresponding parameter, for example name of the file you want to work with.

[parameter] - This means that the parameter may be used, but if you needn't it, you can simply skip this one, and type rest of the example. (Don't type the characters "[" and "]" !)

(ON|OFF) - This tell you that you need to type "ON", or "OFF" (not both).

name - Name of a file or directory. Names can be up to 8 characters long, and may contain upper case letters, numbers, and the "_" character. Any character, which is not allowed in the name, will terminate the whole specification of a file.

You can also use wild cards: "*" means that the rest of name may be any, and "?" means that a single character may be any. This is not possible while creating a new file or directory! Most of commands will work with the first selected file while using wild cards, but a few commands will work with every selected files (see descriptions for single commands).

For example "A*" will select files with any name which begins with "A", "?A*" is for every files with "A" at second place, and "TE?T" will select for example "TEST", "TEXT" etc.

ext - extension part of filename. It may be up to three characters long, and it shows type of the file. The extension may be any, but it is recommended to use standard extensions which are easy to understand, and some programs supports them automatically. Standard extensions are for example "COM" for a machine code program, "BAT" for a batch file, "BAS" for a program in Atari Basic, "TXT" for a text file, "DOC" for a documentation file (in fact a text file), "PIC" for a picture, and so on. It is recommended to use no extensions for directories.

Dn: - The specification of a disk drive, where "n" is number of the drive. You can also type simply "D:" for drive 1.

device: - Specification of device, which is not a disk drive. For example "E:" for screen editor, "C:" for tape recorder, "P:" for printer etc.

filename - Full name of a file - it is "name[.ext]". While writing a file, you can add /A after filename to add the new file at the end of old file with the same name. (Without /A the new file will replace the old one.) A filename with wild cards in it may be also called as "filter".

path - Specification of directory, in which the wanted file is. There are two kinds of path - Absolute: ">[name][>name]...", and relative: "[<][<]...[name][>name]...". See explanation later in this manual for more details.

file - This means full specification of file to work with. The syntax is "[Dn:][path>]filename".

Every numbers are entered as hexadecimal numbers (without of the $ identification). File positions are up to 6 digits long, while memory addresses and sector numbers are up to 4 digits long.

System Architecture

BW-DOS is divided into two parts:

1. Resident part (the DOS itself) is loaded while booting, and it stay in the memory to provide basic functions of the system. (Like every other DOSes.) The resident part come from a file with extension "DOS", and it contains two large routines: Command Processor (CP), and File Management System (FMS). CP is the program which is waiting for your commands, executing some of them, and loading nonresident programs for the rest. FMS is used by every other programs (including the CP) to get access to files on the disk; it is organizing every data written on the disk.

2. Nonresident part is divided into several small programs made to provide more functions. Such a program must be loaded from disk every time you want to use it, so you need to keep these programs on every disks where you need to use them, or on a disk in another drive (or in a RamDisk).

There are three kinds of commands for the command processor:

1. Internal commands are fully executed by the CP, so you can use these on any disk, and you can return to Basic (or other program if possible) later, without of data loss.

2. External commands are executed by additional programs loaded from disk, so you need to have them on the disk you are working with, or on a disk in another drive (or RamDisk) - in this case you must type "Dn:" before the command. Every use of an external command results in data loss from user memory, so you can't restart any program without off loading it from disk again. Most of other programs (like programming languages, editors etc.) may be loaded in the same way as external commands - simply by typing name of the program. (If there is no "COM" extension, you must type the real extension too!)

3. Third kind of BW-DOS commands are resident commands. They are in fact external commands with extension SYS, but there is a important difference: When you'll start a resident command, a part of the command's program will stay in memory - like resident part of the DOS do - and it will provide more "internal" functions. Note that this will increase the MEMLO value, so it's possible that some programs will not work then (because of collision in memory). Every resident commands of BW-DOS may be also removed from memory by starting them again with parameter "OFF" (for example „RAMDISK.SYS OFF"). But remember that you can only remove the last installed resident command! (So, if you have for example installed JIFFY, and then RAMDISK.SYS, and you want to remove JIFFY.SYS later - you must remove RAMDISK.SYS first, then you can remove JIFFY.SYS, and then install RAMDISK.SYS again.) Once you've installed a resident command, which have no remove ability (a non BW-DOS one), you can remove every resident commands installed at that moment only by booting the system up again.

Commands may use parameters (for example filenames); you need to type parameters after the command name (on the same line), separated by the space character. The line with a CP command may be up to 64 characters long.

Every BW-DOS commands may be aborted by pressing the <ESC> key while the program is waiting for an answer from the user, or while displaying something in an endless loop.

IMPORTANT: Never change a disk if there is a file open on it. That is while a batch file from the disk is working, while the HardCopy function is sending output into a file on the disk, while you are editing a file etc. In addition, while a file is open in write or update mode, don't press <RESET> and don't turn the computer off.

Disks and Directories

Every disk under BW-DOS have its name (Volume), and main directory. There may be almost unlimited number of other directories, each placed in another one like a file.

Viewed from a directory, every directories which are in it are "subdirectories", and viewed from a subdirectory, the directory in which it is may be called "parent directory".

BW-DOS directories may contain up to 1424 files or directories, but it is recommended to keep this number less than 100, because the access to long directories is slow. Besides of this, while working with SpartaDOS, only SpartaDOS X (version 4.x) is able to use BW-DOS directories in full - other versions will work only with the first 126 files (the rest will be invisible).

Working directory is directory, where all the files will be searched when no path is specified, or where the relative path have its begin. For each disk drive there is a working directory, which is independent of another drives. Every disk change, <RESET>, jump through DOSINI vector, or booting up will set the main directory as working.

You can change current drive number by typing the new "Dn:". This drive number will be used when no "Dn:" is specified. Note that this only works in the command processor

The path is used to specify in which directory the wanted file (or directory) is. Path is simply a list of directories you need to go through from working directory to the target directory. Single names (including filename after the path) are separated by the ">" character. There are two kinds of path: Absolute and relative. Absolute path begins with a single ">" character, and it begins in main directory, while relative path have no ">" at its begin, and it begins in working directory. In addition, at the begin of relative path there can be any number of "<" characters - each one tell BW-DOS to go into parent directory.

For example ">DATA>TEXTS>filename" means that the file is in directory "TEXTS", which is in directory "DATA" in main directory. "MUSIC>filename" means that the file is in directory "MUSIC", which is in the working directory. "<BASIC>filename" means that the file is in directory "BASIC", which is in the same directory, where the working directory is.

Special directory is "DOS" in the main directory. Every files for reading (or programs for starting), which are not in working directory, are searched in the "DOS" directory too. This works only when no path is specified!

It is recommended to place external commands, the DOS itself, and other commonly used "read only" files in this directory to provide a simple access never mind the working directory.

Besides off this, it's a very good idea to create separate directories for different file types, such as Basic programs, texts etc. (When you'll save everything to the main directory, it'll become very long soon - the access will be slow, and the listing will be almost unreadable.)

Batch Files and Hard Copy

This syntax will start a batch file is -filename. Batch files are in fact a text files, which contains a list of commands to be executed. Standard extension for a batch file is BAT. When you start a batch file, the computer will read commands not from the keyboard, but from the batch file. This works not only in the command processor, it works in most of other programs too (for example Atari Basic). There are of course exceptions - for example the menu program, several text editors etc.

You can use batch files in two ways:

1. When you need to execute several "time eating" programs, put their list into a batch file, start it, and then you can for example go to the lunch - the computer will work on its own.

2. You can use batch files to write a simple programs for some tasks, for example to install Ramdisk and copy selected external commands to it etc.

A batch file named STARTUP.BAT (in main directory, or in the directory "DOS") will be executed every time you boot the system. It is good to place commands for system configuration in such a file. Since BW-DOS supports no CONFIG.SYS file, you need to put a name of such a program into STARTUP.BAT.

For example, you want following happen after booting the system: Install Ramdisk, turn the internal Basic off, switch to directory ATMAS, and run ATMAS II macro assembler, which is in main directory under the name ATMAS.COM. The file STARTUP.BAT could look like this:

```
RAMDISK.SYS 8
BASIC OFF
CWD >ATMAS>
```

Every error message in the command processor will stop any batch file to prevent problems. The batch file will be also aborted by starting another one.

You can add lines with comments into a batch file. Such a line must begin with a ";" character.

If you are a SpartaDOS user, you probably noticed that there is no XDIV command. That's because the HATABS table is in its original state while no batch file or HardCopy function is active. That means that you can start programs - which needs the XDIV in SpartaDOS - directly.

Batch files and the HardCopy function works in the following way: A small routine is inserted between CIO and the handler of "E:" device by changing the "HATABS" table (at $31A). This routine checks all Editor input/output, if is it going through IOCB 0. If it is OK, another handler will be used to get data from a Batch file, or to send data to the HardCopy device.

Every batch files are accessed through IOCB 5, and the HardCopy function uses IOCB 4. These IOCBs are modified to look like a closed IOCB, so a CLOSE command (done by many programs automatically - for example the Basic) may not cause any problems.

Date and Time

Every files and directories have its date and time in BW-DOS - it is the date/time when they were created. To provide this information while creating a file, there is a clock in BW-DOS.

Because of little memory available for the DOS, the BW-DOS's clock is not a real one - it only keeps the date/time information, without of changing it. This means, that every files will get the same date/time till you enter a new value.

If you want to have a real clock in BW-DOS, you need to use some of the resident commands that are designed for this purpose.

CLOCK.SYS is a software clock for BW-DOS. If you've entered the correct date/time (with DATE and TIME commands), then every files and directories will get the real time when they are created.

While installed, this clock will get the date/time from old clock. While removed (or a re-boot), the new date/time will be lost. This clock is based on the VBI interrupt, so it is possible that a few programs will stop it. It works on both PAL and NTSC machines.

There are also driver for other hardware clocks:

- IDE plus 2 - IDEP2TD.SYS
- Ultimate 1MB - ULTD.SYS
- Side 3 - S3TD.SYS
- RTime 8 - RTIME8.SYS

There is the command APETIME that reads time and date view SIO command from all APE (Atari Peripheral Emulator) compatible devices like SIO2USB and FujiNet. This command can be used to set the date and time for the CLOCK.SYS driver. In STARTUP.BAT the command sequence looks like this:

```
CLOCK.SYS ON
APETIME
```

Installation

This chapter explains how to install BW-DOS on floppy disk or hard disk. Prerequisites are that you have booted from the BW-DOS disk and that the destination medium is insert in drive 2 or hard disk is mounted to drive 3.

For a preliminary configuration you can start the batch file CONFIG.BAT with the following command:

```
-CONFIG
```

This batch file will ask you if you want to install a Ramdisk and ask you for installation of a clock. Chose the clock that fits your configuration. Questions are answered with <Y> or <N>. The batch file can be canceled any time with <ESC>.

If you want to read a short description for a command, e.g. FORMAT just type:

```
MAN FORMAT
```

If the manuals are not on the actual drive then you specify the drive:

```
MAN D1:FORMAT
```

To install BW-DOS an a new disk there are four steps:

1. Format and initialize disk
2. Copy files to disk
3. Make disk bootable
4. Adapt Start-Up file

1. Format and Initialize Disk

The command FORMAT is used to format and initialize floppy disks. The examples assume that the new floppy is inserted in drive 2.

```
FORMAT
```

To initialize a hard disk partition the command HDINIT is used. Both commands ask for the necessary information and can be started without any command line parameters. The command FDISK can be used to create and mount APT (Atari Partition Table) compatible partitions, e.g. Side1/2/3 or IDEplus. Make sure that the new partition is assigned to drive 3.

```
FDISK
HDINIT
```

Detailed documentation for FDISK and APT can be found here:
https://atari8.co.uk/apt/documentation/

2. Copy Files to Disk

If the new disk is formatted and initialized, directories created and files copied. For the examples assume that in drive 1 is the BW-DOS disk and the destination disk is the actual drive. To make the new disk the actual drive use command D2: for floppy disks or D3: for hard disk.

Create a new folder DOS:

```
MD DOS
```

All commands of BW-DOS require ca. 130kB of disk space. If your destination does not provide so much free disk space, you can just copy the commands you need. There is always the program file (extension COM or SYS) and the manual file without extension. You can also skip all manual files but then command MAN will not work.

Copy all BW-DOS commands to the new DOS folder:

```
D1:COPY D1:>DOS>*.* DOS>
```

Copy Start-Up file to the new drive:

```
D1:COPY D1:>*.*
```

Note: Use the D1: qualifier for the COPY command to load the command from drive 1 because up until now there are no files (commands) on the destination disk.

3. Make Disk Bootable

The file to be booted is `DOS>XBW14.DOS`. This is the executable file to be started at boot phase. If all files copied to the new disk, the command BOOT can be used to make this disk bootable.

Use the following command for floppy disk in drive 2:

```
BOOT D2:>DOS>XBW14.DOS
```

Use the following command for hard disk assigned to drive 3:

```
BOOT D3:>DOS>XBW14.DOS
```

4. Adapt Start-Up File

The last step is to adapt the Start-Up file STARTUP.BAT.

Use the command EDIT to change the start-up file:

```
EDIT STARTUP.BAT
```

If you want to switch off the key click and install driver for the clock of the Ultimate 1MB then STARTUP.BAT file could like this:

```
POKE 02DB FF
ULTD.SYS
```

If you want to switch off the key click and install the software clock and setting date and time then STARTUP.BAT file could look like this:

```
POKE 02DB FF
CLOCK ON
DATE
TIME
```

The statements DATE and TIME will ask for new date and time for the software clock.

The Menu

```
MENU [Dn:][/S(E|N|S|D|0)][/B(Y|N)][/Q(Y|N)]
```

The external menu command is a comfortable tool for operations with many selected files, for copying between two disks with only one disk drive, and for copying files from Atari DOS 2 format to BW-DOS and back.

Every commands in the menu - excepting these ones for subdirectories - works on Atari DOS 2 compatible disks too. This function is only supported by the menu program, so it is NOT available in any other part of BW-DOS. Following Atari DOS 2 compatible formats are supported: Single density (DOS 2.0, DOS 2.5, BiboDOS, and others), Enhanced (Medium) density (DOS 2.5), and Double density (BiboDOS, and others). Note that in Atari DOS 2 alike systems there are different conditions for filenames, so when you are converting files to Atari DOS 2, make sure that there is no "_" character, and that the first character of the name is not a number.

While working with the menu, you'll see disk directory on your screen. It shows the same information as the listing provided by DIR command in the command processor. If you are working with an Atari DOS 2 disk, there is no date/time information, and the length of files is not exact. Above the directory there is the current drive number and path. A little arrows on the top and bottom edge of the directory shows that the directory continues outside your screen. The menu program can work with max. 255 files in a directory.

The cursor is on the left edge; you can move it with <up> and <down> keys, and with <SPACE> you can select or deselect single files. Selected files are shown in inverse video, and when you select a command later, it'll be executed for every selected files. (When no files are selected, it'll be executed for a file pointed by the cursor.)

At the bottom of your screen there is a command menu with another cursor in it. Use keys <left> and <right> to move this cursor, and press <RETURN> to start the selected function. (Description of single functions later.) To select a command quickly, you can move the cursor where you want with a single keystroke - see the descriptions for correct keys.

There is an input line between directory listing, and the command menu. Here the menu will ask you for everything what's necessary. While selecting from a menu here, you can use the <left>, <right>, and <RETURN> keys, or simply hit the first letter of the wanted selection. While entering a filename or path, you can use the keys <DELETE>, <CONTROL>+<left>, <CONTROL>+<right>, <CONTROL>+<DELETE>, <CONTROL>+<INSERT>, and <RETURN> in the same way as in standard system editor; in addition you can use the <CONTROL>+<CLEAR> combination to clear the line and start typing again.

`New disk <F>`

> This function allows you to select another drive number to work with; you'll see the working directory of selected drive. Use this function when you want to work with another drive, or when you've changed the disk in drive (you'll see main directory in this case).
>
> The current path to working directory (before executing this command) will be saved. Later, this path will be used as a default while copying files to the corresponding drive.

`Disk info <Z>`

> With this function you can look at a little info about the disk. It gives the same listing as CHKDSK command in the command processor.

`Sub dir <T>`

This function allows you to go into a subdirectory. You need to place the cursor to a subdirectory first.

`Up dir <Y>`

This is the function to go into the parent directory. The cursor will be placed on the directory you came from.

`Make dir <N>`

With this command you can create new subdirectories.

`Del. dir <D>`

The "Del. dir" command allows you to delete a subdirectory. It is only possible when the subdirectory is empty.

`Copy <C>`

This is the main function for copying files using the menu. You'll be asked for destination drive and path, and then every selected files will be copied here. If you was working with the destination drive before the last use of "New Disk" function, or if you was copying to that drive already, then you'll get a default path to confirm or edit. The destination path may be up to 231 characters long (it scrolls when necessary).

If you select the same drive number, you'll be asked if is it the same disk - when your answer will be "No", then the menu program will ask you for changing source and destination disks, so you can copy from one disk to another with only one drive. When you are copying to the same disk, you must copy into other directory!

In the case that a file with the same name already exists in the target directory, a little menu appears. You can overwrite the old file, skip the file, or abort the copying. This feature may be turned off when necessary (see "Setup").

`Erase <E>`

This is the function for erasing files. Protected files may not be erased.

`Rename <R>`

This function gives new names to files. You can select more files, and then enter a filter as a new name to rename more files by single command (see description of "RENAME" command).

`Protect <P>, Unprotect <U>`

Functions for protecting and unprotecting files or directories. A protected file may not be changed or erased.

`View text <V>`

This function will copy the file pointed by cursor to the screen. It is the best way how to read small documentation files while working with the menu.

`Make text <I>`

With this function you can make small documentation files. Press <CONTROL>+<3> when you've finished the whole text. (This command will provide the same function as "COPY E: file" in the command processor.)

`Ainit <A>`

This is the command for formatting Atari DOS 2 compatible disks. You'll be asked for drive number and density, and after the last confirm the disk will be formatted. The format created by this command may be used under several DOS 2 alike systems, but BW-DOS may not use it (only the menu can).

Warning: Formatting will erase every files on the disk, never mind if they are protected or not.

`Exit <Q>`

This is the selection to quit the menu, and return back to command processor.

Setup <S>

With this function you can select the order, in which the files will appear on your screen. Files may be sorted by its extension and name, by its name and extension, by its length, by the date and time they were created, or you can leave the directory in the same order as on the disk (this may be useful when the directory is already sorted or it can save a few seconds when the directory is very long).

Second question allows you to turn on/off the sound signal that accompanies errors, and other important messages.

The last question is for turning on/off the menu-selection before overwriting a file - see COPY).

All the selections may be also done from the command processor, using parameters while starting the menu. "Dn:" specifies drive number the menu will start with, and the rest is similar to the "Setup" function ("/S" is sorting of directories, "/B" is the bell function, and "/Q" controls the question before overwriting a file.)

`Select <INSERT>, Deselect <DELETE>`

These functions will select or deselect files specified by entered filter.

Commands

Some of external commands described in this chapter are marked as "low memory". This means that the commands are loading into the area of $480-$6FF, and they are not using the memory areas of $80-$FF, and MEMLO-MEMHI. Such a commands allows you to work with data or machine code programs in the main memory without of problems.

List of internal and external commands:

`APETIME`

This external command reads the APE hardware real-time-clock and sets date and time. It works with all SIO clocks compatible with the APE protocol like SIO2PC and FujiNet. It can be used together with the resident CLOCK.SYS. First install CLOCK.SYS and than use APETIME to set BW-DOS clock.

`BASIC ON|OFF`

This internal command switches the internal Basic (XL/XE only) on or off. With Basic off you have more free memory, and several programs works in this mode only. This command works only with no cartridge installed, and it'll always erase data from user memory (Basic program).

`BLOAD file address [length]`

This is the external command for loading data files (non DOS loadable) into memory. When no length is specified, the whole file will be loaded.

This command uses low memory.

`BOOT file`

With this external command you can specify a file to be loaded, while the system is booted up (when you turn the computer on with the disk in drive 1). Commonly it is a DOS file, but other files may be booted too - see the conditions below.

The common use of this command is installing the DOS on a disk that was previously formatted with no DOS, or with an older version. To install a DOS on a disk, you should copy the X*.DOS file (the DOS directory is recommended), and then use the BOOT command to install it.

The booting process is done by a little loader, which is in sectors 1-3 on every BW-DOS (or SpartaDOS 2.x and later) disks. The file which is booted must be in the DOS loadable format, and it must contain a RUN address. If there are not at least two bytes between end of file, and end of physical data sector (that is in Basic: "IF 128*INT((LNGTH+127)/128)-LNGTH<2"), then two zero bytes must be added as an EOF mark. The booted file may not be loaded into the area of $2E00-$317F, where is the loader itself. Note that the loader is little different from the one found on SpartaDOS disks: If you are booting from a SpartaDOS disk, then no segments in the file - excepting the first one - may begin with the $FFFF header. Besides of this, the SpartaDOS loader is not fully compatible with the XF 551 disk drive.

When you'll erase the file specified to be booted, an attempt to boot such a disk gives unpredictable result.

`CAR`

This internal command will jump to a program in cartridge, or to internal Basic (if enabled - only XL/XE). If you've not used an external command since you leaved the cartridge, your data (Basic program) will not be lost.

CD [Dn:]path

This internal command will set the directory specified by path as working directory.

CHKDSK [Dn:]

This external command will display info about the disk. It contains volume name, two internal numbers (sequential and random - used for disk identification), version number of the SpartaDOS file system, size of sectors, and total/free disk capacity in bytes. The listing looks:

```
     Volume: BWDOS14  53 FF
    Version: 2.0
Sector size: 128 bytes
   Capacity:    92160 bytes
 Free space:    67200 bytes
```

CHVOL [Dn:]

This external command will change the volume name of disk. Dn: or if not specified the current drive

CLEANUP Dn: [/P]

This program is a part of SpartaDOS toolkit, which was released as ShareWare. With this program you can fix several problems on BW-DOS disks.

Option P switches on the printer for report.

The format of BW-DOS RamDisks may be reported as bad by the CLEANUP program. Unfortunately, this program contains a bug. To prevent problems, you should press <RESET> immediately after leaving CLEANUP (under both BW-DOS and SpartaDOS).

This command was developed for SpartaDOS (copyright by ICD/FTe).

COLD

This external command executes a cold start. Before cold start hard copy and batch processing is stopped.

COMP file1 file2 [maxdiff]

This external command compares two files. The maximum differences to be displayed can be specified. The default for maximum differences is 16. If the files are nor equal then the program terminates with error 255.

COPY [Dn:][path>][filename] [Dn:][path>][filename[/A]]

This is the main command for copying files; it'll copy file(s) specified by the first parameter to file(s) specified by the second one. You can use any I/O device instead of Dn:
(for example "COPY K: P:"; most of these devices will ignore any filenames).

When copying disk files, you can use wild cards in both the filenames to copy a set of files, and save them with another names (see REN command). When no filename is specified, every files in directory will be copied (first parameter), or the files will get their original names (second parameter). To copy a file to itself will destroy the file!

Examples:

```
COPY >DOS> D8:
```

Copy the whole directory DOS to drive 8 (RamDisk).

`COPY E: file`

Create a new text file. Press <CONTROL>+<3> when you've entered the whole contents for new file.

`COPY file P:`

Send file(s) to the printer.

`CUT file1 file2 position [length]`

This external command saves a specified part of file1 as file2. If no length is specified, it'll save the maximum length - that is between position and the end of file.

It is possible - for example - to divide DOS loadable files into single segments using OFFLOAD and CUT, to cut off binary file headers from DOS loadable files with on segment:

```
CUT file1 file2 6
```

Command requires 48k main memory.

`DATE`

This external command will display the current date from BW-DOS's clock, and allows you to enter a new value. Press <RETURN> to leave it unchanged.

Command requires 48k main memory.

`DEL file`

This internal command erases files. When you'll use wildcards, you can erase several files with one command. Protected files and directories are invisible for this command (see PROT and UNPROT).

`DIR [Dn:][path>][filename]`

This internal command show contents of the specified directory. When no filename is specified, every files will be listed. This listing starts with the volume name of the disk, and name of the listed directory. It shows for each file filename, length, and date/time when the file was created. Directories are indicated by "<DIR>" listed instead of length. At the end of listing you'll see number of free sectors on the disk. The listing looks like this:

```
Volume:    TESTS
Directory: MAIN

DOS          <DIR>  11-06-94 16:01
TEST1    BAS   1231 11-06-94 16:32
   573 FREE SECTORS
```

`DIRS [Dn:][path>][filename]`

This internal command provides the same function as "DIR", but the listing is in Atari DOS 2 style now. Directories are indicated by "DIR" in inverse video instead of extension. Note that length of files is in sectors now (one sector contains 128 bytes in Single or Medium density, or 256 bytes in Double density). The file in fact occupies a few sectors more; they are used for internal informations for the FMS. The star before filename means that the file is protected. Every numbers in this listing (file length and number of free sectors) are only 3 digits long; any number greater than 999 will be shown as "999". This listing looks like this:

```
* DOS     DIR 000
```

```
* TEST1   BAS 010
573 FREE SECTORS
```

DISASS file [position]

This external command will list contents of specified file as an assembler listing. With a position specified, it'll start from this position. Press <CONTROL>+<1> to stop the listing and continue, or <ESC> to abort this command.

If you want to display a program code from memory, you need to save it to a disk file first. For example, to display a program code from memory area $8000-$9000, you need to type:

```
SAVE D1:TEMP.DAT 8000 9000
DISASS D1:TEMP.DAT 6
```

DUMP file [position [length]] [/A]

This external command will display contents of specified file as hexadecimal and ATASCII listing. With the /A parameter it'll change non printable characters (0-$1F and above $80) to printable ones - use this for printing the listing (see the "PRINT" command).

If a position is specified, the listing will start from this position. With the length parameter only specified number of bytes are displayed.

Press <CONTROL>+<1> to stop the listing and for continue, or <ESC> to abort this command.

Command requires 48k main memory.

EDIT [file]

This external command is a text file editor. If no file is specified a new one is created and the name is specified at save. Press <HELP> for in program help.

This program is developed by FJC (se https://atari8.co.uk).

FA src[.ext] [obj[.ext]] [/LSEY]

Fast Assembler 1.8 17-03-2023, an assembler for SpartaDOS X and BW-DOS.

src - ATASCII source file
obj - target binary object file

The default extensions (ext) are .ASM and .OBJ. The default obj name is the src name.

/L list combined source

/S print summary, e.g. block and label count, memory
$code-$stat_alloc-$last dyn_alloc-last

/E sets compile errors to system errors starting from 224($E0)

/Y dumps symbole table to file obj.SYM

For more information see https://github.com/HolgerJanz/FastAssembler/

FORMAT [Dn:]

This external command formats disks, and installs BW-DOS. Formatting must be the first thing done on a new disk. The drive number specified as a parameter is used for searching DOS files,

not for formatting! The program will ask you for necessary input, and then format the disk. This program may not be controlled from a batch file. Ramdisks may not be formatted in this way.

Warning: Formatting will erase all data on the disk - never mind if they are protected or not - and it cannot be unerase anymore. The only protection against this is to place the write protect tab on your disk.

The first question is which DOS version you wish to install on the formatted disk. You'll get a little menu with every DOS files found on the disk (files "X*.DOS" in directory "DOS", and in the main directory) - select the correct one, or press <N> for none. It is possible to install any version of BW-DOS, or any disk based version of SpartaDOS 2.x and later.

Next questions are for drive number, density, volume name, and if use the XF 551 high speed format or not. Note: You can only select a density, which is supported by your drive. That is for Atari 810 - Single; Atari 1050 - Single or Medium; expanded Atari 1050 - Single, Medium, or Double; XF 551 - Single, Medium, Double, or Double sided Double density; other drive - see manual for the drive.

Now insert the disk you want to format, make sure that it is the disk you want to format, and press <RETURN> to format it. If the disk was formatted in the same density already, and you want to clear it only, press <B> - in this case the FORMAT program will skip physical formatting.

After formatting it'll create BW-DOS disk structures, and if you've selected a DOS, it'll create the directory "DOS" and install the DOS file there.

The last question is if you want to format another disk.

Command requires 48k main memory.

`GETTD`

This external command will display the current date and time from BW-DOS's clock.

`HDINIT`

This external command writes an initial file system to a mounted hard disk. The hard disk must be mounted to drive 3 or 4. All necessary parameter are ask via dialog by the command. <ESC> will abort the command at any time.

Command requires 48k main memory.

This command was developed for SpartaDOS (copyright by ICD/FTe).

`HEXEDIT file [position]`

This external command allows you to edit contents of a file in hexadecimal mode. With a position specified, the editing will start from this position.

It'll display position in the file, the old value, and you can enter new value (hexadecimal), or just press <RETURN> to leave it unchanged. Press <ESC> to quit this command.

Note: The only legal way to quit HEXEDIT is the <ESC> key. Because the edited file is open all the time you are working with HEXEDIT, it may be corrupted by pressing <RESET> or turning the computer off before HEXEDIT finish its work.

It is possible to set the position greater than the length of file - in this case no old value will be displayed; once you'll enter any value, the file will get a new length. When you set the position far after the end of file, the part of file between old and new data will not be physically stored (non-

allocated sectors) - it'll get a physical sector at the time of first access to such a place. Files with non-allocated sectors may not be read or copied in a normal way, because allocating a sector in a read-only mode isn't allowed.

IF/ELSE/ENDIF

These external commands used in batch files only. It will ask the user, which block of commands should be executed.

The syntax is:

```
IF
the question
commands...
ELSE
commands...
ENDIF
```

It will display the line with question, and wait for an answer <Y> or <N>. If the answer is <Y>, then all the commands between IF and ELSE are executed, if the answer is <N>, then all commands between ELSE and ENDIF are executed. The <ESC> key will stop the batch file. If there are no commands for the answer <N> then the ELSE command can be omitted.

It is possible to make much more complicated structures by starting other batch files inside an IF or ELSE block - but there is no way to return to the first batch file. You can only start it from begin.

LOAD file

This internal command will simply load a DOS loadable file (program), but the program is not started after loading. Any INIT or RUN address will be ignored.

MD [Dn:][path>]name

This internal command will make a new directory specified by "name". Note that you can't make a directory with the same name as a file in the same directory.

MDUMP address [length] [/A]

This external command displays contents of memory as hexadecimal and ATASCII listing. With the /A parameter it'll change non printable characters (0-$1F and above $80) to printable ones - use this while printing the listing (see the PRINT command).

This command uses low memory.

MEM

This external command returns memory areas containing memory and OS ROM info.
First the accessible main memory is showed e.g., XL with Basic off is 62 kB (64 kB minus 2 kB hardware register $d000-$d7ff). Free memory shows the memory between MEMLO and MEMTOP. Memory info area $6a, $02e4-$02e8 containing RAMTOP, RAMSIZ, MEMTOP, and MEMLO. Extended memory info contains kind (Rambo, Compy, or Axlon compatible), size in kilo byte and .
OS ROM info area $c000-$c001 and $ffee-$fff9 contains:

```
$c000-1 CHK1:  Checksum 1, two bytes (LSB/MSB)
$ffee   DAT:  Revision date D1 and D2 (four-bit BCD)
$ffef   DAT:  Revision date M1 and M2
$fff0   DAT:  Revision date Y1 and Y2
$fff1   XLO:  Option byte; should read 1 for the 1200XL, other XL/XE reads 2
$fff2-6 PN:   Part number in the form AANNNNNN
```

```
$fff7    REV:  Revision number
$fff8-9 CHK2:  Checksum 2, two bytes (LSB/MSB)
```

Example output depending on your machine:

```
RAM Info:
  Main: 62 kB Free 40259 bytes
  RAMSIZ=C000 RAMTOP=C000
  MEMLO =1EDD MEMTOP=BC1F
  Compy:  256 kB Banks: 16

ROM Info:
  REV=02 DAT=100583 XLO=02
  PN=4242000001 CHK1=9211 CHK2=6C8C
```

MEMCLEAR [vv]

This external command clears memory between MEMLO and MEMTOP with $vv. Default value 00. Can be used for testing and analyzing. Example:

```
MEMCLEAR FF
```

MEMEDIT address

This external command allows you to edit contents of memory. Use this only if you know exactly, what you are doing. It is possible to cause the system to be corrupt, or even to lock up with this command.

It'll display position, the old value, and you can enter new value (hexadecimal), or just press <RETURN> to leave it unchanged. Press <ESC> to quit this command.

MENU [Dn:][/S(E|N|S|D|0)][/B(Y|N)][/Q(Y|N)]

This external command is a comfortable tool for operations with many selected files, for copying between two disks with only one disk drive, and for copying files from Atari DOS 2 format to BW-DOS and back (see chapter The Menu).

"Dn:" specifies drive number the menu will start with, and the rest is similar to the "Setup" function: “/S" is sorting of directories, "/B" is the bell function, and "/Q" controls the question before overwriting a file.)

Command requires 48k main memory.

MOVE file [Dn:][path]

This external command allows you to move a file or subdirectory into another directory on the same disk quickly. The greatest difference between COPY and MOVE is that MOVE removes the file from source directory, and is much faster. It also doesn't need any free sectors on the disk, because it only moves the filename into an other directory - without of copying the file itself.

Command requires 48k main memory.

`OFFLOAD file [offset] [/Q|L]`

This external command displays the structure of a DOS loadable file. The starting and ending address, and the file position of the data block (after the header) will be displayed for each segment.

With the /L parameter it'll also load the file into memory, and with /Q it'll ask you for each segment. When an offset is specified, it will be added to loading address of each segment.

This command works like the commands marked as "low memory" only while loading a file. It uses a memory area starting at $6000.

`PAUSE`

This external command is used in batch files only. It will wait for the <RETURN> key. The <ESC> key will stop the batch file. The PAUSE command can be used to ask for disk change in batch files. Do not change the disk on which the running batch file is read from.

Command requires 48k main memory.

`PERCOM [Dn:]`

This external command reads the Percom block from supporting drives. If no drive is specified the current drive is read. If the drive does not support Percom blocks, error 139 (Device NAK) is returned. See chapter Disk Forkmat

---

`POKE aaaa vv[vv] ...`

This external command writes a byte vv or a word vvvv to the memory address aaaa. All parameters are hexadecimals. It can be used to change system variables. Examples:

`POKE 02C6 00`
Writes value $00 to address $2C7 COLOR2 (sets background color to black).

`POKE 02E7 2050`
Writes value $50 to address $02e7 and value $20 to address $02e7+1 (sets MEMLO to address $2050).

A list of address and value can be passed for several pokes:

```
POKE 02C6 00 02E7 2050
```

`PRINT [device:|file]`

This external command is used for output control. With parameter it will start the HardCopy function, without it will stop this function. HardCopy function allows you to copy all the screen output from almost every programs to a specified device or file. It doesn't work with programs, which are not using CIO for screen output. Originally this command redirect output from programs to a printer but you can redirect the output to any file/device.

`PROT file`

This internal command is protecting files. Protected files cannot be changed or deleted. With wildcards you can protect a set of files. This command can also protect directories.

`RD [Dn:][path>]name`

This internal command deletes directories. You can only delete an empty directory - if there are any files or directories in it, you need to remove them first. Never try to delete the working directory.

`REN file filename`

This internal command allows you to rename a file specified by "file" parameter to new name "filename". You can use wild cards in both the names to rename a set of files, and to leave specified characters in the name unchanged. (For example when you have files TEST and BEST, after the command "`RENAME ?EST ????S`" you'll get files TESTS and BESTS.)

Use wildcards only if you know exactly, what are you doing! When you'll change two filenames in a directory to be the same, you can't work with the second one anymore, and the only thing you can do is to erase such a "double file".

You can't rename directories with this command. See RENDIR.

`RENDIR directory newname`

This external command allows you to rename a directory specified by "directory" parameter to new name "newname".

This command was delivered for SpartaDOS (copyright by ICD/FTe).

`RUN [address]`

This internal command will start a program at given address. With no parameter it'll restart the last loaded program, or the last program started with this command.

This command is there for advanced users and programmers. RUN with a wrong address results in a system lock up, and it may even cause a data loss.

The RUN command may be used by everyone for a few purposes:

RUN - Restart the last loaded program (but first look into the manual for your program to know if is it possible, or not);

RUN E471 - This will jump to the Self Test program (XL/XE only), or to the menu of Q-MEG operating system (if installed).

`SAVE file start end [/A]`

This will save the memory area specified by start and end addresses as a DOS loadable file. With the /A parameter it will add another segment to an existing file.

The difference between /A as a part of filename or as a separate parameter is that the separate /A will not place another $FFFF header into the file.

This command uses low memory.

`SECOPY Dn:[filename] Dm:[filename]`

This external program copies sectors from disk or file to disk or file. Works only with 128 Bytes sector disks. Disks are not analyzed or formatted. Sectors are copied until either read or write error occurs. Examples:

SECOPY D1: D8:

Copies sector from D1: to D8:

SECOPY D8: D4:>BACKUP>RAMDISK.DSK
Copies sectors from D8: to file

SECOPY D4:>BACKUP>RAMDISK.DSK D8:
Copies file to sectors of D8:

`SORTDIR Dn:[path] [/NTSDX]`

This external command sorts the entries of a specified directory. There are the following sort options:

N : Sort by Filename (default)
T : Sort by File Type
S : Sort by File Size
D : Sort by Creation Date
X : Reverse Sort

The sort order is the order the command DIR and DIRS will display entries.

Command requires 48k main memory.

This command was developed for SpartaDOS (copyright by ICD/FTe).

`TIME`

This external command will display the current time from BW-DOS's clock, and allows you to enter a new value. Press <RETURN> to leave it unchanged.

Command requires 48k main memory.

`TYPE file`

This internal command shows to content of text files (for example batch files like STARTUP.BAT). Use the keys <CTRL>-<1> to stop and continue the output..

`UNPROT file`

This internal command is unprotecting files (see PROT). This command can unprotect directories too.

`VERIFY ON|OFF`

This external command switches the write with verify option on or off. The default is OFF. With option ON every sector read after writing and data are compared. This results in a very slow writing. It can be useful while using bad quality disks.

Command requires 48k main memory.

Resident Commands

List of resident commands (.SYS):

`CLOCK.SYS ON|OFF`

This resident command is a software clock for BW-DOS. If you've entered the correct date/time (with "DATE" and "TIME" commands), then every files and directories will get the real time when they are created.

While installed, this clock will get the date/time from old clock; while removed (or a re-boot), the new date/time will be lost. This clock is based on the VBI interrupt, so it is possible that a few programs will stop it. It works on both PAL and NTSC machines.

The parameter ON installs the clock and OFF removes the drive, if it was the last driver loaded.

Command requires 48k main memory.

`IDEP2TD.SYS`

This command installs a driver for a hardware real-time-clock. Such a device gives the correct date/time information to the system without of a need to re-enter the information after each re-boot. It works with the PBI (Parallel Bus Interface) device IDE plus 2 by JZ & KMK (see http://drac030.krap.pl/en-kmkjz-pliki.php)

`RAMDISK.SYS (n[E][F] [file])|OFF`

This resident command sets up an emulated disk drive using extended RAM (up to 1MB, PORTB and Axlon compatible). This means that you can use your extended memory like a disk drive - excepting formatting and booting. All files on Ramdisk is lost if you turn off the computer, or if you use a program, that uses extended memory for other purposes.

The "n" parameter is the number of ramdisk drive.

The "F" option will cause the ramdisk to be formatted. Without option "F" it will only format if the previously existing format is invalid. Without of "F", you can mostly re-install the driver (even after a re-boot) without of destroying the contents of your Ramdisk.

The "E" option will cause that it will not use 130XE compatible banks (only for machines with more than 128kB memory), so it'll not collide with 130XE software. This option is ignored for Axlon or if a configuration file is used.

PORTB is check first and if there are no PORTB banks then Axlon banks are check. Afterwards the bank count and kind are displayed.

If the ramdisk collides with your software and you know that there are some memory banks left, you can use the configuration file. This file is a binary file and may be between 1 and 64 bytes long. It contains the values which must be stored at PORTB $D301 or Axlon $CFFF register to get different banks in the area of $4000-$7FFF. Sizes between 16kB and 1024kB are possible. The driver checks the values before activating the ramdisk. The command HEXEDIT can be used to create and edit a binary configuration file. The command DUMP can be used to display a configuration file.

PORTB configuration files contain the value for PORTB register, e.g. for an Atari 130XE (4 banks) the binary file can contain the following hexadecimals value:

EF EB E7 E3

Axlon configuration files contain the bank number from 01 to FF. To define a ramdisk of 64k with the first 4 banks the binary file contain the following hexadecimals values:

01 02 03 04

There are three example files provided for the PORTB RAMBO 256k extensions. This extension is special because the main memory is also banked.

`R256D192.BNK`
This file contains all banks except the banks for the main memory. 12 banks are used for the ramdisk. The file is 12 Bytes long

`R256D128.BNK`
This file contains all banks except the banks for the main memory and the 130XE banks. 8 banks are used for the ramdisk. The file is 8 Bytes long.

`R256D208.BNK`
This file contains all banks except the banks for the main memory but the bank of RAM under the OS ROM. 13 banks are used for the ramdisk. You cannot use any programs that used the RAM under the OS ROM with this ramdisk. The file is 13 Bytes long.

Example for PORTB Rambo 256k:
`RAMDISK.SYS 8 R256D192.BNK`

To reformat an existing Ramdisk, use the same configuration and option „F“. If you want to change the drive number then start the same command only with a different drive number. In this case the ramdisk will be preserved.

Note that the resident part of Ramdisk driver uses a part of extended RAM for itself, so any collision in the extended RAM may cause a lock up of SIO operations.

RAMDISK.SYS OFF removes the driver, if it was the last driver loaded.

`RTIME8.SYS [/R]`

This command installs a driver for a hardware real-time-clock. Such a device gives the correct date/time information to the system without of a need to re-enter the information after each re-boot. It works with the R-Time-8 cartridge by ICD (or other compatible one).

With the /R parameter a reduced version will be installed. It allows only reading from the clock, setting a new date/time isn't possible in this mode. This option saves your memory-space.

`S3TD.SYS`

This command installs a driver for a hardware real-time-clock. Such a device gives the correct date/time information to the system without of a need to re-enter the information after each re-boot. It works with the Side3 cartridge designed by Sebastian Bartkowicz (‘Candle O’Sin’) and with software by FJC (see https://atari8.co.uk/apt/toolkit/).

`ULTD.SYS`

This command installs a driver for a hardware real-time-clock. Such a device gives the correct date/time information to the system without of a need to re-enter the information after each re-boot. It works with the Ultimate 1MB extension designed by Sebastian Bartkowicz (‘Candle O’Sin’) and with software by FJC (see https://atari8.co.uk/apt/toolkit/).

Functions in Basic

This chapter shows, how to use the functions of BW-DOS in your own programs written in Basic. In other programming languages the commands (mostly) are not very different.

This chapter is there for programmers, so don't worry, when you don't understand.

```
OPEN #iocb,aux1,aux2,"Dn:[path>]filename"
```

This command will open specified IOCB for access to the specified file on disk. The value of "aux1" may be:

4 - Reading from the file.

6 - Reading the directory listing. When the aux2 value is greater than 127, the listing will be in the same format as provided by DIR command in the command processor, with aux2 less than 128 it'll be in the format of DIRS command. You may not open more than one directory listing at the same time.

8 - Writing to the file. If the file exists already, it'll be replaced by the new file. When "/A" is added to the filename, it'll work in the same way as the mode 9.

9 - Append, the new data written to the file will be added at the end of an existing file. When the file doesn't exist, it'll be created first.

12 - Update. Data may be read from and written to the file as necessary.

When you'll add 16 to the aux" value, you'll get direct access to the directory specified by path, the filename will be ignored. With 32 added to aux1, you'll get the direct access to the subdirectory specified by filename. (See the next chapter for information on the internal directory format.) This doesn't work with the mode 6.

When you are opening a directory for direct access in mode 8, 9, or 12, and the directory doesn't exist, it'll provide the MD function first. The position in directory data will be set to 23 (after the first block) in every modes excepting the mode 9 - if you want to read/write the first block, then you must use POINT.

Warning: Use the direct access to directory only when you know exactly, what are you doing! With this mode it's possible to destroy the whole directory structure on a disk by a single OPEN command.

```
GET #iocb,variable
PUT #iocb,variable
INPUT #iocb,variable
PRINT #iocb,variable
```

These commands are sending data to the specified IOCB, or receiving data from the IOCB. Note: It is much faster to send data as one large block. This may be done by CIO commands 7 and 11 in the assembler, or by commands BGET and BPUT in Turbo Basic.

```
CLOSE #iocb
```

This is the command for closing files. Every file must be closed after use.

```
X=POS:Y=0:POINT #iocb,X,Y
```

or

```
Y=16*iocb:X=INT(POS/65536):POKE 846+Y,X
POS=POS-65536*X X=INT(POS/256):POKE 845+Y,X
POKE 844+Y,POS-256*X
XIO 37,#iocb,PEEK(842+Y),PEEK(843+Y),"Dn:"
```

In this way you can set the position in a file (the POINT function); the position is simply serial number of the byte in file. When the file is opened in mode 8, 9, or 12, then you can set the position after the end of file, too.

The first method works only for positions less than 32768, while the second method may be used for any position up to 8 megabytes. The IOCB must be opened first.

```
Y=16*iocb:XIO 39,#iocb,PEEK(842+Y),PEEK(843+Y),“Dn:"
LNGTH=PEEK(844+Y)+256*PEEK(845+Y)+65536*PEEK(846+Y)
```

This command will provide the information how long is the file, for which the IOCB is opened.

```
NOTE #iocb,X,Y:POS=X+65536*Y
```

With this command you'll get the current position in a file, for which the IOCB is opened. (This may be done in the same way as the previous command too, the XIO command code is 38.)

```
XIO 32,#iocb,0,0,"Dn:[path>]filename filename"
```

This command will provide the same function as the "RENAME" command in the CP. The space character between old and new filename may be replaced by any character, which is not allowed in the filename - for example "," . The IOCB must be closed before this operation.

Following commands will provide the same functions as several CP commands - as listed. The IOCB used for such a command must be closed first.

```
XIO 33,#iocb,0,0,"Dn:[path>]filename"    = the DEL command
XIO 35,#iocb,0,0,"Dn:[path>]filename"    = the PROT command
XIO 36,#iocb,0,0,"Dn:[path>]filename"    = the UNPROT command
XIO 40,#iocb,4,0,"Dn:[path>]filename"    = load and run a program
XIO 40,#iocb,4,128,“Dn:[path>]filename“  = the LOAD command
XIO 42,#iocb,0,0,“Dn:[path>]name“        = the MD command
XIO 43,#iocb,0,0,“Dn:[path>]name“        = the RD command
XIO 44,#iocb,0,0,"Dn:path"               = the CD command
```

The last function may not be (easily) used from Basic. The settings for CIO are:

```
ICCMD = 47
ICBAL/H = address of the string "Dn:"
ICBLL/H = address of the output buffer
```

It will provide the information similar to the "CHKDSK" command; the result in output buffer will be always 17 bytes long:

```
0.  : Version of disk format
1   : Size of sectors. (0 means 256.)
2,3 : Number of every sectors
4,5 : Number of free sectors
6-13: Volume name
14  : Sequential number
15. : Random number
16. : Always 0 (reserved for compatibility with SpartaDOS).
```

Disk Format

BW-DOS uses the same disk format as SpartaDOS. There are four kinds of sectors: Boot sectors, Bitmaps, Sector maps, and Data sectors.

Boot sectors are sectors 1, 2, and 3 on every disks. These sectors are always 128 bytes long, and contains a small program, which is started while you turn the computer on - this program is loading DOS (or other file specified by the BOOT command). The Boot loader is not the same on BW-DOS and SpartaDOS disks, but the function of both the loaders is almost identical.

In sector 1, there is an important table, which gives to the system several informations about the whole disk (the numbers are positions in the sector 1):

7: Always $80. This byte is used as an identification of SpartaDOS compatible disk. When this value is different, you'll get the Error 148 while accessing such a disk.

9: Sector, where is the first Sector map of the main directory. (2 bytes)

11: Number of every sectors on the disk. (2 bytes)

13: Number of free sectors. (2 bytes)

15: Number of Bitmaps.

16: Sector used for the first Bitmap. (2 bytes)

18: The starting sector-number for searching a free sector for a file; commonly it is the last allocated sector. This pointer speeds up write operations, because the DOS need not to search the whole Bitmap for a free sector. In addition, it'll leave a few sectors on first tracks of the disk unused to provide space for directories. Such a space will be used for files only when there is no other free sector on the disk (this state is indicated by the value of the pointer equal to maximum sector -1). While erasing a file or directory, this pointer will be set to maximum sector /16 to allow use of any sectors that are free after the erase operation.

20: Starting sector for allocating directories - see the previous explanation. While erasing a file or directory, this pointer will be set to 2.

22: Volume name of the disk. (8 bytes) This name - together with the sequential and random numbers - is used for identification of the disk. (The DOS bases its decision if the sectors in buffers are correct on this.)

30: Number of tracks; when the bit D7 is set, then the drive is double sided.

31: Size of sectors (excepting Boot sectors). Possible values are 128, or 0 (for 256 bytes/ sector).

32: Version of disk format. BW-DOS (and every SpartaDOS versions 2.x and later) uses the same format marked as $20 in this byte. Older disks ($11 - created by SpartaDOS 1.x) may not be used under BW-DOS.

33-37: Reserved for different SpartaDOS versions.

38: Sequential number of the disk. This number is incremented every time you're making any changes on the disk using BW-DOS functions.

39: Random number of the disk. This number is created while formatting the disk. 40: Sector, where is the first Sector map of file specified for booting. (2 bytes)

42-47: Reserved for different SpartaDOS versions.

The Bitmap contains the information, which sectors are free, and which are used. Each byte holds the information for eight sectors - the highest bit D7 is for the first, and the lowest bit D0 for the last of these eight sectors. The bit set to 1 means, that the sector is free. The first byte contains the information for sectors 0-7, second byte for 8- 15 etc.

When more than one sector is necessary for the map, it is stored on the disk as a compact block (it may be for example in sectors 4, 5, and 6).

Sector maps contains information, which sectors are used for a file. There are physical numbers of sectors (2 bytes each) in a Sector map.

The first number in every Sector map (in fact two bytes!) points to the next Sector map of the file. Zero indicates the last map.

Second number points to the previous map of the file; zero indicates the first map.

The rest of the map contains numbers of data sectors used for the file.

Directories are in fact special files; they may be directly accessed by adding 16 or 32 to the "aux1" value while opening the file. The directories contains information about every files and subdirectories, 23 bytes each. (The numbers are positions in this - 23 bytes long - block.)

0: The status. Single bits are used:
- D0 -The file is protected.
- D3 - Block in directory is used.
- D4 - The file is deleted (block in directory is free).
- D5 - Flag for a subdirectory.

The status equal to zero means the end of directory.
1: Sector, where is the first Sector map of the file.
3: Length of the file (3 bytes - not for a subdirectory).
6: Name and extension (11 bytes).
17: Date and time in the same format as "DATER" and "TIMER".

The first block is different; it contains an information about the directory itself. When you want to read or write this block using BW-DOS, you must use the POINT function - every OPEN in direct mode will set the file position to 23. The information in the first block is this:

1: Sector, where is the first map of parent directory.
3: Length of the directory in bytes (3 bytes).
6: Name of the directory (8 bytes).

Some disk drive support Percom blocks (see command PERCOM). The Percom block is a configuration block. It is a set of 12 bytes within the memory of the disk control microprocessor. The bytes in the Percom block have the following meaning:

1	Number of Tracks
2	Step Rate (values have no universal meaning)
3-4	Sectors per Track (byte 3 is high byte; byte 4 is low byte)
5	Number of Sides or Heads (0=one head, 1=two heads)
6	Density (0=FM/Single, 4=MFM/Double)
7-8	Bytes per Sector (byte 7 is high byte; byte 8 is low byte)
9	Drive Selected
10	Serial Rate Control (values have no universal meaning)
11-12	Miscellaneous (reserved)

Page 7 and COMTAB

In the memory page 7 there is a table, that shows version of the DOS, and allows machine code programs to use a few special functions:

$700: BW_SPARTA
There is always the character "S" to provide the best compatibility with SpartaDOS. Many programs will determine by this address, if they can use advanced SpartaDOS (BW-DOS) functions.

$701: BW_SVERS
There is always the number $32 to provide the best compatibility with SpartaDOS 3.2.

$702: BW_SSVERS
There is always the number $00 to provide the best compatibility with SpartaDOS 3.2.

$703: BW_DOS
Two bytes. There are the letters "BW" when any version of BW-DOS is installed.

$705: BW_VERS
BW-DOS version. $10 is for version 1.0x, $14 is for version 1.4x etc.

$706: BW_GETTD
Address of the GETTD routine. This routine reads the date and time from BW-DOS's clock, and store it into DATER and TIMER in the COMTAB.

$708: BW_SETTD
Address of the SETTD routine. This routine reads the date and time from DATER and TIMER in the COMTAB, and store it into BW-DOS's clock.

$70A: BW_CONVDC
Address of CONVDC routine. This routine will convert a 24-bit binary number from DECIN in the COMTAB to the 8 characters long ASCII text in the DECOUT.

$70C: BW_BBSIO
There is an instruction, which jumps to the SIO routine address in BW_LSIO in the COMTAB. This is provided for compatibility with BiboDOS. Any changes done to this address may not change function of BW-DOS.

$70F: BW_FAIL
Address for error handling and jump to DOSVEC. This routine will never return. The error code must be passed in register A. Error handling includes termination of batch processing and output of error number. It can be used for error handling of SIO calls e.g.:

```
jsr BW_SIO
bpl no_error
tya
jmp (BW_FAIL)
```

This address is only available in BW-DOS as from version 1.4 Check BW_DOS and BW_VERS before use.

The COMTAB table allows machine code programs to get parameters from the command processor, and to use many other features of BW-DOS. Address of this table is stored in the DOSVEC vector (at $0A). Numbers in the following description are relative positions in the table.

With BW-DOS version 1.4 the COMTAB can also be absolutely addressed in page 7. Check BW_DOS for ‚BW' and BW_VERS for greater equal $14 before using absolute access to COMTAB. This is not compatible with SpartaDOS.

$711: BW_DWARM
COMTAB-21: This value will be stored to the address WARMST (at $08) before starting a cartridge. It provides the information if the memory was changed by an external command $00, or not $FF.

$713: BW_DECOUT
COMTAB-19: This is an output buffer for the BW_CONVDC routine. It contains up to 8 characters long decimal representation of the number from BW_DECIN. Spaces may be added at the begin. This address is compatible with SpartaDOS X only.

$71B: BW_SIO
COMTAB-11: Jump to BW_LSIO (see BW_LSIO).

$71C: BW_LSIO
COMTAB-10: There is the address of main BW-DOS's SIO routine. This routine is used by BW-DOS for every disk operations. It has the same input and output like the standard SIO in ROM ($E459). After booting this address points to $E459. It may be changed by commands like RAMDISK.SYS.

$71E: BW_ECHOFLG
COMTAB-8: This is the index into HATABS for the hard copy function. There is the value $FF if the hard copy function is not active. It is possible to get information, if the hard copy function is active reading this byte.

$71F: BW_BATFLG
COMTAB-7: BATFLG - It is the same as BW_ECHOFLG. This flag is used for batch files.

$720: BW_DECIN
COMTAB-6: DECIN - Input for the BW_CONVDC routine (3 bytes). This address is compatible with SpartaDOS X only.

$724: BW_WRTCMD
COMTAB-2: WRTCMD - There is the SIO command used for every write operations. It is "P" for writing without verify, or "W" for writing with verify. Any other value may cause big troubles. It is recommended to use the SIO command from this byte for all write operations in every programs.

$726: BW_COMTAB
COMTAB+0: COMTAB - At this point, there is an instruction for starting the command processor. You can use the instruction JMP ($0A). It will always start a command processor, DUP menu, or something like that.

$729: BW_CRNAME
COMTAB+3: CRNAME - Here is an instruction, which jumps to the CRNAME routine. This routine will get the next parameter from command line in LBUF, add Dn: if none was specified, and place the result in COMFNAM. N=0 if parameter was fetched, N=1 no more parameter.

$72C: BW_DIVIO
COMTAB+6: DIVIO - Here is the address of DIVIO routine. This routine may be used to start a batch file or the hard copy function. Before starting this routine, the complete filename ("device:" or "Dn:[path>]filename") must be placed in COMFNAM, and the Y register must be set to 0 for the hard copy function, or 1 for a batch file. This routine will return any errors in the Y register and N flag in the same way as CIO or SIO. That's different from SpartaDOS 3.x.

$72E: BW_XDIVIO
COMTAB+8: XDIVIO - There is the address of XDIVIO routine. This routine may be used to stop a batch file or the hard copy function. Before starting this routine, the Y register must be 0 for the hard copy function, or 1 for a batch file.

$730: BW_BUFOFF
COMTAB+10: BUFOFF - This is the position of next character in LBUF for the CRNAME routine.

$733: BW_DATER / $736: BW_TIMER
COMTAB+13: DATER / +16: TIMER - This is the main variable for date and time (3+3 bytes). It is in the format "day, month, year; hour, min., sec." This variable is used by GETTD and SETTD routines, and while creating a file.

$739: BW_ODATER / $73C: BW_OTIMER
COMTAB+19: ODATER / **COMTAB+22**: OTIMER - This variable is in the same format as DATER and TIMER; it is used while copying files (see TDOVER).

$73F: BW_TDOVER
COMATB+25: TDOVER - This flag determines if the new created file will get the current date/time from BW-DOS's clock ($00), or the date/time from ODATER and OTIMER ($FF). It is used especially while copying files to carry the date/time information. Don't forget to clear this flag after use.

$740: BW_TRUN
COMATB+26: TRUN - Here is the RUN address from the last file loaded by the command processor (or with the XIO 40 command).

$744: BW_SMEMLO
COMTAB+30: SMEMLO - The address of first byte in memory after the BW-DOS itself. If the MEMLO value is different, then there are resident commands between SMEMLO and MEMLO.

$746: BW_INCMND
COMTAB+32: INCMND - This address determines, if you are working with the command processor or with a cartridge, to provide the correct function of the <RESET> key. If there is $FF, then the <RESET> key will jump to the command processor.

$747: BW_COMFNAM
COMTAB+33: COMFNAM - This is a buffer for complete device and filename specification. It is used for output from the CRNAME routine, and as input for the DIVIO routine. It is 28 bytes long. When reading parameters from the command line with the CRNAME routine, this buffer will always start with "Dn:" - if you are looking for other parameters than filenames, you must start at COMFNAM+3.

$763: BW_RUNLOC
COMTAB+61: RUNLOC - There is stored the address, which will be used by the RUN command without parameter. This address is changed by loading a program, or by RUN with a parameter.

$765: BW_LBUF
COMTAB+63: LBUF - This is the buffer for command line (and for DIR, DIRS, and TYPE commands). There will be stored the whole command line before executing it. This buffer is 64 bytes long.

If you want to program compatible to SpartaDOS then you have to use the following pattern. In this example pattern the command line is parsed for one parameter with CRNAME and the SIO vector is used. Furthermore XDIVIO is used in error handling. The pattern could look like this:

```
…
* set crname
            lda DOSVEC
            clc
            adc #$03 ; CRNAME = COMTAB+3
            sta crname+1
            lda DOSVEC+1
            adc #$00
            sta crname+2
* set sio
            lda DOSVEC
            sec
            sbc #11 ; SIO = COMTAB-11
            sta lsio+1
            lda DOSVEC+1
            sbc #$00
            sta lsio+2
…
            jsr crname
…
            jsr sio
            tya
            bmi error
…
* call CRNAME
crname      jmp $ffff
* call SIO
sio         jmp $ffff
* error handling
error       ; print error message in any way
…
* stop batch
            ldy #$08 ; XDIVIO = COMTAB+8
            lda (DOSVEC),Y
            sta xdivio+1
            iny
            lda (DOSVEC),Y
            sta xdivio+2
            ldy #$01
xdivio      jsr $ffff
* back to DOS
            jmp (DOSVEC)
```

The same pattern using absolute COMTAB addressing could look like this:

```
…
BW_CRNAME   equ $729
BW_SIO      equ $71b
BW_FAIL     equ $70f
…
            jsr BW_CRNAME
…
            jsr BW_SIO
            bpl sio_ok
            tya
            jmp (BW_FAIL)
sio_ok
…
```

The jump to the BW_FAIL vector will print the standard error message with error code in register A, abort a running batch processing, and at the end a jump back to the command processor.

Note that this is not compatible with SpartaDOS. Programs using the second pattern will only run with BW-DOS 1.4. For this you should check BW_DOS for ‚BW' and BW_VERS for greater equal $14.

Errors

This chapter gives a list of Error codes (in decimal and hexadecimal numbers), and their descriptions. Every Error codes less than 128 have its origin in other programs (for example Basic), so there is no connection between such Errors and BW-DOS.

Not every listed errors have its origin in BW-DOS. There are also errors caused by CIO, SIO, and other device drivers. (For example screen, tape recorder etc.)

The first two codes are not errors - BW-DOS returns these codes when everything is OK. You'll encounter these codes only while programing CIO calls in assembler or something like that.

001($01) OK - This code is returned by all device handlers when no error occurs.

003($03) Last Byte read - This code is returned by BW-DOS instead of the code 1 ("OK"), in the case that the last byte of the file was read. Next reading will return the error-code 136 ("End of file") - except that you're working in the "update" mode (aux1=12). This code will not be returned in the directory-listing mode (aux1=6). This code have no connection with the error 3 returned by Basic or other programs.

128 $80 BREAK - The user pressed the <BREAK> key during an input/output operation.

129 $81 IOCB already open

130 $82 Non existent device - The specified device does not exist. Please check if the corresponding handler is installed. If you are accessing a disk file, try to add "Dn:" before the filename - this is always necessary while accessing a file from your own programs.

131 $83 IOCB write only

132 $84 Invalid CIO command

133 $85 IOCB not open

134 $86 Invalid IOCB number

135 $87 IOCB read only

136 $88 End of file - This is not a real error. This code indicates, that there are no more bytes left to read in the file.

137 $89 Truncated record - The line in a file is longer than buffer for it. This may occur while reading an incorrect type of file.

138 $8A Timeout - The device is not connected correctly, it is turned off, or it doesn't exist. Check drive number while accessing a disk drive.

139 $8B Device NAK - The device didn't understand. This may occur for example while using programs for modified drives with a standard disk drive.

140 $8C Serial frame error - This error may be caused by tape recorder, or by damaged input/output device, computer, or cables.

141 $8D Out of screen - The position of cursor is out of screen.

142 $8E Serial bus overrun - These errors may be caused by tape recorder, or by damaged input/output device, computer, or cables.

143 $8F Checksum error - These errors may be caused by tape recorder, or by damaged input/output device, computer, or cables.

144 $90 Device done error - The device can't execute the command. The most common reasons are: Write protect tab placed on the disk while writing; a non formatted disk, or no disk in the drive; bad sector on the disk.

145 $91 Illegal screen mode - The selected screen mode doesn't exist.

146 $92 Function not implemented - This error may be caused by actions like reading data from the printer or writing to the keyboard.

147 $93 Insufficient RAM - The selected screen mode needs more memory than available.

148 $94 Invalid disk format - The disk is not a BW-DOS compatible disk. You can only use disks created with the FORMAT command in BW-DOS, or with SpartaDOS 2.0 and later.

150 $96 Directory not found - A directory specified in the path doesn't exist.

151 $97 File already exists - A new opened file have the same name as a subdirectory in the same directory, or a new created subdirectory have the same name as another file/subdirectory in the same directory.

152 $98 Not a binary file - The file is not a DOS loadable program-file.

156 $9C Parameter or Cartridge error - The CAR command didn't found any cartridge, or a wrong parameter was used.

160 $A0 Drive number error - The specified number of disk drive is not supported. BW-DOS only supports drives 1, 2, 3, 4, and 8.

161 $A1 Too many open files - The number of open files at the same time is too high. BW-DOS may open 5 files at the same time, but only one of them may be a directory listing (aux1=6).

162 $A2 Disk full - There are no more sectors left on the disk, so you need to use anoter one. The part of file, which was written before this error, is not completed, so it must be erased.

163 $A3 Invalid wild card in file name - A wild card was used while creating a new file or directory.

164 $A4 File is protected - The file may not be changed, because it is protected. See commands PROT and UNPROT.

165 $A5 Illegal file name - Please check the file specification. File name contains illegal characters.

166 $A6 Illegal file position - The file position given to the POINT function is greater than 8 megabytes, or it is greater than length of file when it is open in a read only mode (aux1=4). This error may also occur when you try to copy a file, which contains non-allocated sectors. While working with Atari DOS 2 compatible disks using the MENU program, this error indicates that the file number in a sector is incorrect. This may be

caused by a non-compatible disk format, or by a corrupt disk structure.

167 $A7 Directory not empty - The directory specified in the RD command is not empty, so it cannot be deleted.

168 $A8 Command not implemented - BW-DOS didn't understand the command code. Please check the command code for CIO, or the XIO command. This error may also occur while trying to use a few special functions of SpartaDOS, which are not supported by BW-DOS.

169 $A9 Directory full - There is no more space for a new file in directory. BW-DOS directories may contain up to 1424 files, but it is much better to divide such a large number of files into several subdirectories. Atari DOS 2 compatible disks (supported by the MENU program only) may contain up to 64 files.

170 $AA File not found - The specified file or directory doesn't exist. This error may also occur while trying to erase a locked file or subdirectory, or to rename a subdirectory.

Changes from Version 1.3 to 1.4

- Needs only 16kB (not yet for all external commands)
- Starts always command processor at boot. It does not automatically starts a cartridge. Use command CAR to run cartridge.
- Sets left margin always to 0
- Command CREDIR is renamed MD
- Command CWD is renamed CD
- Command DELDIR is renamed RD
- Command ERASE is renamed DEL
- Command PROTECT is renamed LOCK
- Command RENAME is renamed REN
- Command UNPROTECT is renamed UNLOCK
- New internal command MAN
- Additional commands and driver by FJC, and copyright by ICD, and FTe
- Absolute address COMTAB (see chapter Page 7 and COMTAB)
- Change of sector allocation to improve performance at common overwrite cases
- RAMDISK.SYS now supports Rambo 256k with main banking