

NAME

bbcim - BBC computer file conversion utility

SYNOPSIS

bbcim [**options**] [**file(s)**] ...

DESCRIPTION

bbcim is a file conversion utility to help in transferring files and for use with emulators, like creating disk images (Acorn DFS and other formats), adding files to them, extracting files from them, minimise the size of these images etc.

USE

bbcim option [file [{filelist}]] (any extension is allowed for disk images)

In the following listing of options, diskoptions are:

[-type DFS|W62|DDOS|ADFS|HDFS] [-bytes <no>] [-sectors <no>] [-tracks <no>] [-sides <no>] i.e. they identify the exact disk type (except for non-standard or protected types); the default is DFS, 40 tracks (or whatever it really is, as the DFS disk image stores the disk size in total number of sectors), single sided, 256 bytes/sector (standard), 10 sectors/track (this latter number is only of interest for interleaving/splitting double sided disk images, and varies from 10 for DFS, 16 for ADFS and some double density DFS'es such as Solidisk DDFS, to 18 for Opus DDOS).

Two special cases are for filenames ending in .ssd and .dsd which have been used (robert Schmidt) to identify single sided and double sided DFS disks (which are the most common type of disk image).

Options are:

-c[R] [<diskoptions>] <disk> [-side <side>]

show catalogue (files in image). R is for hierarchical filing system images, recursively showing all files from the root or given directory.

-e[s|d,@,b,r] [<diskoptions>] <disk> [-side <side>] [{list}]

extracts files. Without d,s or @ the option, -e extracts files from a disk image 'imagefile' with filenames such as 'imagefile.D.DATA' but acorn DFS root dirs are removed, so '\$.!BOOT' is extracted to 'imagefile.!BOOT'. With every 'bare' file, an info file is created containing in particular the load- and execution addresses (i.e. the archive format).

The suboptions produce the following results:

d extracts the files in a directory named after the disk image (appending ".d"): eg. 'imagefile.d/\$.ELITE'.

s gives short filenames (no leading disk image name nor is a directory created to put the files in). Note that **d** and **s** are mutually exclusive.

@ keeps the Acorn DFS root directory \$, so \$.ELITE doesn't become ELITE.

b only the bare files are created, and no info files (with file attributes).

r NOT IMPLEMENTED: remove the extracted files from the disk image.

When a list of file names is given, only the files in the list are extracted from the disk image.

-a[b,r] [<diskoptions>] <disk> [-side <side>] {list}

add files (in archive format) to the disk image: Automatically skips '.inf' files so bbcim -a <disk> * (on unix) gives no problems. If the disk image didn't exist a new one will be created. The option 'r' removes the archive files after adding them to the disk image. Side can be 0 or 1 (for double sided disk images) *The option 'b' adds files without .inf file* (setting load/exec addresses to 0).

- d** [**<diskoptions>**] **<disk>** [**-side <side>**] **{list}**
 delete the list of files from the disk image. NOTE: files without a BBC (Acorn DFS) directory are assumed to have the root directory \$, eg. ELITE is assumed to be \$.ELITE
- 40** [**<diskoptions>**] **<disk>**
 expand a disk image to 40 tracks (if current size is smaller than 40 tracks). This changes the disk image size, but more importantly also the DFS filing system size (the number with the number of sectors that's stored in the disk image). So shrinking an 80 track image to 40 track is possible by first using -min, then -40 (if the files fit on 40 tracks).
- 80** [**<diskoptions>**] **<disk>**
 expand a disk image to 80 tracks (if current size is smaller than 80 tracks). This changes the disk image size, but more importantly also the DFS filing system size (see above).
- max** [**<diskoptions>**] **<disk>**
 expand a single or double sided DFS disk image to almost 256K per side. This changes the disk image size, but more importantly also the DFS filing system size (see above).
- min** [**<diskoptions>**] **<disk>**
 minimise a disk image (equal to *COMPACT and cutting off after the last used sector). In case of double sided images, both sides are minimised.
- crc** [**<diskoptions>**] **<disk>** [**-side <side>**]
 calculates crc's for all the files in the disk image.
- new** [**<diskoptions>**] **<disk>**
 makes a new (empty) disk image (for Acorn DFS: 512 bytes, bootoption EXEC).
- boot** [**<diskoptions>**] **<disk>** [**-side <side>**] **<bootoption>**
 Sets the bootoption of the disk image: none, LOAD, RUN, EXEC.
- interss** [**<diskoptions 0>**] **<disk0>** [**<diskoptions 1>**] **<disk1>** **<disk2>**
 interleaves two single sided disk images <disk0> and <disk1> to <disk2>.
- splitds** [**<diskoptions>**] **<disk>**
 splits an interleaved disk image <disk> to <disk>.0 and <disk>.1.
- ddos_dfs** [**<diskoptions>**] **<disk>**
 splits a single sided DDOS disk image into several DFS disk images. Eg. myfiles > myfiles.A,..., myfiles.H.
- w62_dfs** [**<diskoptions>**] **<disk>**
 splits a single sided watford 62 file disk image into two DFS disk images (one image if there are less than 32 files on the image) <disk>.1 and <disk>.2.
- id** **<disk>**
 tries to identify a disk image (type/sides).
- s[@]** **<file>**
 splits a text file with file information (catalogue) into .inf files. Lines not starting with a BBC (Acorn DFS) file name (i.e. dir.name) are skipped. This can be used for example to make archive files from an xbeeb directory file (__CATALOG__).
- In DFS use, one is usually in directoy '\$' (the root dir), and the current directory is not displayed. This means, it's useful to remove the root dir '\$' in host file names. If you don't want this, use option '@' which keeps the root directory \$.
- inf_clean** **{list}**
 renames archive/info files to standard format (as produced with '#' in -e (XXX update this as '#'
 was removed XXX)), add's crc's if not found, checks crc's if found.

- inf_crc {list}**
check crc's of the given archive files.
- V** print bbcim version.
- H** gives general help (command overview).
- help <option>**
Gives an extract of the man page pertaining to option <option>.

Disk and boot type are set as follows:

<disk options> is :

[-type <DFS | W62 | DDOS | ADFS | HDFS>] [-bytes <no>] [-sectors <no>] [-tracks <no>]
[-sides <no>]

and **<boot option> is :**

[none | LOAD | RUN | EXEC]

CURRENT VERSION

1.2

NOTES

Implementation weirdness that should be fixed to conform to Acorn DFS:

For options -d, -e, and -a:

* names are case sensitive.

For options -d, -e:

* wildcards are not possible

For option -a:

* There is no check for existing files with a given name, and none are deleted. All files are appended in the directory so several files in a disk image can have the same name (actually useful to get things back to a beeb, no need to rename, by extracting the files and deleting them one at a time you can get at all the files...).

If in doubt: it's not a bug, it is a feature!

THE STANDARD ARCHIVE FILE & DISK FORMAT (version 0.83)

[N.B. This specification was originally included with bbcim v0.83]

Note: optional arguments are in square brackets, '|' denotes a choice between arguments.

THE ARCHIVE FILE FORMAT:

I.e. the standard format for/of files in the BBC micro archive of software on my website.

[N.B. At the time of writing this down (1997), my BBC archive of software was almost all the software for the BBC micro available on the internet (Robert Schmidt's archive was mostly a mirror of what I put up), and I wanted to get rid of diskimages so you could pick and choose software to put into a disk yourself. So, 'the archive' was my archive of BBC software.]

Complementing a bare BBC file 'ELITE' is the attribute file Locked CRC=XXXX NEXT

ELITEdata'. ELITE is just the name used on the host system (as BBC files often contain characters that other filesystems can't handle such as '/') and \$.ELITE in the attribute file is the real name.

In general we have:

- A BBC file, 'myprog'
- A text file 'myprog.inf' which contains: {<TAPE <tfs_filename> | <dfs_filename>} <load address> <exec address> [<file length>] ['Locked'] CRC=<xxxx> [NEXT <next filename>]

Arguments are separated by spaces.

Cassette names can be used in 2 ways;

1. translated to \$.<cassette_name> (i.e. max 12 chars)
2. TAPE <cassette name>

Note that for complete accuracy, TAPE <cassette name> is preferable as it identifies a file as a TAPE file, where 'Locked' has a different meaning to that for disks, and NEXT can be used (see below).

Programs handling archive format files should be aware of 12 character filenames from tapes (\$.tape-name) and should rename them if necessary.

NEXT gives the next expected filename on the tape, which may be used for CHAIN"" or *RUN as used on cassette systems, or to put the programs back on a tape in the correct order, so a file containing TAPE should contain NEXT too (unless it's the last file). Note that <next filename> used for NEXT must be a host filename, as several files can have the same name on tape!

<load address> and <exec address> are hexadecimal strings (without a language dependant identifier), e.g. FFFF8023.

<file length> is also a hexadecimal string and may be included after <exec address> but is unnecessary.

The file attribute 'Locked' may be abbreviated to 'L'.

After 'CRC=' follows the bare file's CRC, as a hexadecimal string of up to four digits.

The CRC is calculated with the algorithm from the BBC user guide p. 399 (The same algorithm as used in the cassette and rom filing system). It is included for various reasons (to check file conversion software, serial- or disk- file transfer to/from a BBC, etc.). See the appendix for a sample implementation.

Host filenames: It is recommended that files only use ASCII and none of the following characters (due to the way they are used in various operating systems): \ : * ? < > . ` ^ "

THE ARCHIVE DISK FORMAT:

Given a BBC disk image, e.g. 'games', we have an accompanying file ('games.dsk'), which contains the following information:

```
'<disktype> <sides> <tracks>T <sectorspertrack>S <bytespersector>B'
```

e.g. DFS SS 40T 10S 256B.

After this line, any information about the disk image can be placed. This should be ignored by utilities that only need the disk format.

=> Suggestions of useful standard information to put there?

If there's no '.dsk' file, utilities should consider the disk image to be 'DFS SS 80T 10S 256B', i.e. standard Acorn DFS.

Appendix: The CRC algorithm, example C code:

```
int bbcim_data_crc(void *data, int length) {
    int i;
    unsigned char *d = (unsigned char *) data;
    int crc;

    crc = 0;

    for (i=0; i<length; i++) {
        int k;

        crc ^= ((*d++) << 8);
        for(k=0; k<8; k++) {
            if (crc & 32768) crc = (((crc ^ 0x0810) & 32767) << 1)+1;
            else crc <<= 1;
        }
    }

    return crc;
}
```

This is the same algorithm as used in the BBC micro's operating system, in particular in tape blocks. See the BBC microcomputer user guide p. 397.

BUGS

Some areas need testing, especially related to double sided diskimages to see if all works well. If you find a bug, send me a report please.

AUTHOR

Wouter Scholten, email: swhs-web@xs4all.nl

Web page: <http://wouter.bbcmicro.net/>

(previous email addresses:

whs@hccnet.nl,
wouter.scholten@wanadoo.nl,
whs@xs4all.nl,
wouters@cistron.nl)

This man page was created & updated: 1 September 1996, 22 October 1996, 14 November 1996, June 1997, April 2000, Februari 2001, April 2001, May 2001, June 2003, February 2009, April 2009, May 2009, 26 January 2014, 28 July 2016, 17 August 2016