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 diskimages (Acorn DFS and other formats), adding files to them, extracting files from them, minimize the size of these images etc.

USE

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

In the follwing listing of options, diskoptions are: [-DFS|W62|DDOS|ADFS|HDFS] [-SS|DS] (i.e. they identify the exact disk type, the default is DFS & SS (single sided Acorn DFS image)).

Options are:

−e[s|d,#,b,r] [<diskoptions>] <disk> [<adfs_dir>] [{list}]

extracts files. Without the d or s or # the option -e extracts files from a diskimage with filenames like 'imagefile.$.!BOOT'. With every 'bare' file, an info file is created containing load- and execution addresses (archive format).
The suboptions produce the following results:

d extracts the files in a directory named after the diskimage: eg. 'imagefile/$.ELITE'.

s gives short filenames (no leading diskimage name). Note that **d** and **s** are mutually exclusive.

removes the bbc root directory $ and glues other directories to the filename. (name clash is possible but unlikely, eg. E.E and EE, in that case you will be asked whether to rename, overwrite, etc.). Note that I've chosen # instead of $ as this is used as a wildcard character in unix.

b makes -e only produce the bare files, and no info files.

r removes the diskimage after extracting files.

In all cases a catalogue file 'imagefile.cat' is made except when a list of file names is given, in which case:
- only the files in the list are extracted from the disk image
- **r** is ignored (Change this: remove the extracted files from the diskimage?)
- a catalogue is not produced.

-c [<diskoptions>] <disk> [<adfs_dir>]

show catalogue (files in image).

−40 <disk>

expand a single sided DFS diskimage to 40 tracks.

−80 <disk>

expand a single sided DFS diskimage to 80 tracks.

−max <disk>

expand a single sided DFS diskimage to (almost) 256 K.

−min <disk>

minimize a diskimage (equal to *COMPACT and cutting off after last used sector).

–crc [<diskoptions>] <disk> [<adfs_dir>]
calculates crc's for all the files in the diskimage.

–new [<diskoptions>] <disk>
makes a new (empty) diskimage (512 bytes, bootoption EXEC). OBSOLETE (-a creates a new diskimage if needed).

–a[b,r] [<diskoptions>] <disk> [<adfs_dir>] {list}
add files (in archive format) to the diskimage: Automatically skips '.inf' files so bbcim -a disk.bbc * (unix) gives no problems. If the diskimage didn't exist a new one will be created. The option 'r' removes the archive files after adding them to the diskimage. the option 'b' *adds files without .inf file (setting load/exec addresses to 0).*

–d [<diskoptions>] <disk> [<adfs_dir>] {list}
delete the list of files from the diskimage. NOTE: files without a bbcdir are assumed to have the root dir, eg. ELITE is assumed to be $.ELITE

–boot <disk> <bootoption>
Sets the bootoption of the diskimage.

–interss sd|dd <disk0> <disk1> <disk2>
interleaves single sided diskimages <disk0> and <disk1> to <disk2>.

–splitds sd|dd <disk>.2
splits interleaved diskimage <disk>.2 to <disk>.0 & <disk>.1

–ddos2dfs <disk.dd>
splits a single sided DDOS diskimage into several DFS diskimages. Eg. myfiles.dd > myfilesA.bbc,..., myfilesH.bbc.

–w62dfs <disk[.ext]>
splits a single sided watford 62 file diskimage into 2 DFS diskimages (one image if there are less than 32 files on the image) disk.w1 & disk.w2.

–s[#] <file>
splits a text file with file information (catalogue) into .inf files. Lines not starting with a bbc (DFS) file name (i.e. dir.name) are skipped. The option # removes the DFS directory $ and glues others to the file name. This can be used for example to make archive files from an xbeeb directory file (__CATALOG__).

–y {list}
renames archive/info files to standard format (as produced with # in -e), add's crc's if not found, checks crc's if found.

–icrc {list}
check crc's of the given archive files.

–V print version no.

–H help (command overview).

CURRENT VERSION

0.99

NOTES

1. options and files can be entered after starting the programs for all options that do not require a file list.

2. for option -d and -e with a list:

2.1 wildcards are not possible
2.2 names are case sensitive.

3. (see also 2) several files in a diskimage can have the same name when using -a (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!

POSSIBLE FEATURES OF LATER VERSIONS:

-e with list, should suboption 'r' remove the extracted files? (using -d)

-id : identify diskformat (single/doublesided, dfs/ddos). It's not possible to do this in a foolproof way so I'm not sure.

-ren rename, -sect nn: custom disksize (for DDOS)

THE STANDARD ARCHIVE FILE & DISK FORMAT (version 0.83)

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 archive)

Complementing a bare BBC file 'ELITE' is the attribute file Locked CRC=XXXX NEXT ELITE-data'. 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 diskimage, 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 diskimage 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 diskimage to be ’DFS SS 80T 10S 256B’, i.e. standard Acorn DFS.

Appendix: The CRC algorithm, example C code:

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

-- OBSOLETE:
For tape emulation with custom cassette formats (eg. ELITE, FORTRESS), or otherwise protected tapes, the info file must contain: "TAPE PROTECTED tapename NEXT nextfile"

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

Custom tape format (6502em)
The binary file is coded as:
FF 04 : DCD bit goes high (ie data carrier STOPs and interrupt occurs)
FF 00 : DCD bit goes low (ie data carrier STARTs and NO interrupt)
FF FF : FF byte from tape hardware
nn : (not FF) byte from tape hardware The info file must contain 'TAPE name'.
-- END OF OBSOLETE SPECIFICATION

BUGS

Not aware of any problems at the moment. if you find one, send a report.

AUTHOR

Wouter Scholten (wouter.scholten@wanadoo.nl; previous addresses whs@xs4all.nl & wouters@cistron.nl)

This man page was created & updated: 1 sept 1996/22oct 1996/14 nov 1996/jun 1997/apr 2000/feb 2001/april 2001/may 2001

Web page: http://home.wanadoo.nl/~wouter.scholten