[ [main](http://archive.retro-kit.co.uk/bbc.nvg.org/index.php3.html) | [what's new](http://archive.retro-kit.co.uk/bbc.nvg.org/whatsnew.php3.html) | [search](http://archive.retro-kit.co.uk/bbc.nvg.org/archive-search.php3.html) | [software](http://archive.retro-kit.co.uk/bbc.nvg.org/software.php3.html) | [emulators](http://archive.retro-kit.co.uk/bbc.nvg.org/emulators.php3.html) | [archive](http://archive.retro-kit.co.uk/bbc.nvg.org/files.php3.html) | [documentation](http://archive.retro-kit.co.uk/bbc.nvg.org/docs.php3.html) | [links](http://archive.retro-kit.co.uk/bbc.nvg.org/netresource.php3.html#links) ]

*** Standard Archive Format***

***Standard format for storing BBC files on other computers***

by [Mark de Weger](mailto:deweger@cs.utwente.nl).

***Use of the standard format***

BBC files stored on disc have certain attributes that are usually required for these files to be loaded and/or run. These attributes are (on Acorn DFS-systems):

 load address

 execution address

 file length

 locked attribute

(On some other disc filing systems there are more attributes, like read/write permission. These are not considered here to keep the standard format as generally applicable as possible.)

If a BBC file is transferred to a PC (or any other computer) without these attributes (e.g. using Kermit), some required information for loading/running these files is lost. Therefore these attributes need to be transferred to the PC and stored there. The standard format determines the format in which these attributes are stored. In addition it determines the name of a BBC file when stored on a PC (because some legal BBC file names are not e.g. legal DOS file names).

The standard format was first described by Wouter Scholten and is sometimes known under the names "archive format" or "image format".

***The standard format***

The standard format is as follows:

 Every BBC file is stored under a certain name (usually, but not necessarily, its original BBC file name) on the PC. This name may or may not include a directory name. (So the BBC file D.FILE can be stored on the PC under the names FILE or D.FILE, or even as MYNAME.)

 For every BBC file there is an additional file on the PC that contains the BBC file's attributes. This additional file has the same name as the BBC file on the PC plus the extension .INF (e.g. FILE.INF, D.FILE.INF, or MYNAME.INF in the above example).

The syntax of the contents of the .INF file is as follows (attributes in square brackets are optional):

[< filename>] < load address> < exec address> [< length>] [< lock>] [< crc>] [< next>]

where

**< filename>**  
The file name of the BBC file, which may include a directory name. If this is different from the file name as stored on the PC's disk, < filename> overrides it. (The use of this < filename> attribute is that it allows e.g. DOS-users to store a Beeb file with a directory name or illegal characters in the file name. The file name "D.FILE" is an illegal DOS file name. The BBC file "D.FILE" can now be stored on the PC as "FILE" (or any other name), and < filename> in "FILE.INF" be set to "D.FILE".)

**< load address>**  
The load address of the BBC file in hex. The load address cannot be longer than six characters (as on the BBC). Leading zeroes may be given, but are not required, i.e. 0E00 and E00 are both valid.

**< exec address>**   
The execution address of the BBC file. Same requirements as < load address>.

**< length>**  
The length of the BBC file. Same requirements as < load address>.

**< lock>**  
This is "Locked" or "L" (without the parentheses) for locked files and empty otherwise.

**< crc>**  
This is "CRC=XXXX" with XXXX the CRC check of the file, computed with the algorithm present in the Advanced User Guide.

**< next>**  
This is "NEXT < filename2>" (without the parentheses) with < filename2> the name of the file that sequentially follows the current file. (This attribute is useful for storing BBC files from tape--rather than from disc--, because it allows emulators to interpret e.g. CHAIN "" correctly.)

Example of storage of files on PC according to the standard format:

$.ELITE : BBC file $.ELITE stored on PC as $.ELITE

$.ELITE.inf : attribute file for $.ELITE containing the following text:

FF0E00 FF8023 Locked CRC=1234 NEXT $.ELITE2

***Some specialisations***

In order to make the standard format simpler and applicable to all platforms (which is what standards are about), the following specialisations can be made. (These specialisations are only further restrictions of the standard format, not changes to the format. Files stored according to the specialised format thus fully comply with the standard format.) The specialisations have been implemented in BBCXFer.

 The name of a BBC file as stored on the PC should be a valid DOS file name. Preferably the PC file name is obtained as folows as follows:

 the directory name (including the dot) is stripped from the BBC file name

 any character in the BBC file name that is not an allowed character according to DOS is replaced by a \_.

If the conversion results in files with the same names, a file may be given any name, as long as it is unique and a legal DOS file name.  
(This allows BBC files to be stored as valid DOS files. DOS is probably the most restrictive platform in this respect, so allowed DOS file names are also allowed on other platforms. Making the file name of the file stored on the PC resemble the BBC file name is not really required, but just nice.)

 The file name in the .INF file always states the real BBC file name.  
(Making the file name not optional anymore makes the standard format simpler.)

 The file name, the file length and the locked attribute are obligatory and not optional.  
(This makes the standard format simpler.)

 The value of the locked attribute is either "Locked" or empty. The value "L" is not allowed.  
(This makes the standard format simpler.)

 The crc attribute and the next attribute are dropped.  
(This makes the standard format simpler. The crc attribute is also not required, since the sending and receiving programs can compute the crc themselves--the receiving program always has to do so. Moreover, the crc is heavily dependent on the actual algorithm used for computing the crc's--there is no standard for this. The next attribute does allow for emulation of a BBC with a cassette recorder--so that e.g. CHAIN "" is allowed--, but no existing emulator supports this. Moreover, if you're going to transfer software to your Beeb, you're going to store this on disc, aren't you?)

***The specialised standard format***

The above restrictions result in a more specialised standard format. This format is used by BBCXFer for storing files on a PC that are transferred from a BBC. (BBCXFer is able to read Wouter Scholten's more general format when files are transferred from a PC to a BBC.)

The new, specialised, standard format is as follows.

 The name of a BBC file as stored on the PC should be a valid DOS file name. Preferably the PC file name is obtained as follows:

 the directory name (including the dot) is stripped from the BBC file name

 any character in the BBC file name that is not an allowed character according to DOS is replaced by a \_.

If the conversion results in files with the same names, a file may be given any name, as long as it is unique and a legal DOS file name. (This allows BBC files to be stored as valid DOS files. DOS is probably the most restrictive platform in this respect, so allowed DOS file names are also allowed on other platforms. Making the file name of the file stored on the PC resemble the BBC file name is not really required, but just nice.)

 For every BBC file there is an additional file on the PC that contains the BBC file's attributes. This additional file has the same name as the BBC file on the PC plus the extension .INF

The syntax of the contents of the .INF file is as follows:

< filename> < load address> < exec address> < length> < lock>

where

**< filename>**  
The file name of the BBC file, which may include a directory name. If this is different from the actual BBC file name as stored on the PC's disk, < filename> overrides it. (The use of this < filename> attribute is that it allows e.g. DOS-users to store a Beeb file with a directory name or with characters in the file name that are illegal in DOS. The file name "D.FILE" is an illegal DOS file name. The BBC file "D.FILE" can now be stored on the PC as "FILE", and < filename> in "FILE.INF" be set to "D.FILE".)

**< load address>**  
The load address of the BBC file in hex. The load address cannot be longer than six characters (as on the BBC). Leading zeroes may be given, but are not required, i.e. 0E00 and E00 are both valid.

**< exec address>**  
The execution address of the BBC file. Same requirements as < load address>.

**< length>**  
The length of the BBC file. Same requirements as < load address>.

**< lock>**  
This is "Locked" (without the parentheses) for locked files and empty otherwise.

Example of storage of files on PC according to the specialised standard format:

ELITE : bare BBC file $.ELITE, stored on PC as ELITE

ELITE.inf : attribute file for ELITE containing the following text:

$.ELITE FF0E00 FF8023 1230 Locked

OWNNAME : bare BBC file R.ELITE, stored on PC as OWNNAME

OWNNAME.inf : attribute file for OWNNAME containing the following text:

R.ELITE 2000 4300 3010

[Robert Schmidt](http://bbc.nvg.org/~rsc) - *rsc@nvg.org*.