

## **CC2Spriter – Manual**

Version: v2.8.1

Date: Feb. 14th, 2004

Last revision: Jan. 5th, 2006

Author: Mafi

"Close Combat - A Bridge Too Far" (abbreviated CC2, ABTF, CC2-ABTF) was the second game of the CloseCombat-series created by Atomic and presented by Microsoft to the Mac-community. It was also the last game of this series for the MacOS. The series was then continued by SSI/UbiSoft/Mattel Interactive/Destineer for PCs only (up to day CC3, CC4, CC5, CCM, Road to Baghdad) and the last sequel of this series is the USMC training tool "CCM v2" made Summer 2005 by CSO\_SimTek. CC2-ABTF was released in 1997 on a hybrid-CD, running on PCs and under the MacOS 7.5 up to 9.2.2 / MacOS X as well. Later (localized) releases of CC2 were for PCs only.

### **Purpose of this tool:**

- extracting / rebuilding the CC2 sprite files (indicated by the header ID (first four bytes) "SPRI"). Sorry, but Soldier/SoldierB files cannot be rebuilt by this tool,
- extracting / rebuilding the CC3's Terrain file (header ID "SPRI"),
- extracting / rebuilding CC3's vehicle shadow files (\*.nsd / \*.zsd, header ID "SPRI"),
- extracting / rebuilding the CC4/CC5/CCM/RoadToBaghdad Terrain files (header ID "IRPS"), TerrainA.stm (header ID "ALPH") and CC4-RtB-vehicle-shadow \*.spr files (header ID "IRPS" with varying version-ID),
- extracting the main graphical contents out of the CC3/.../RtB files stored in \*.ZFX and \*.AZP archives (includes interpreting the file "FXData.fx", this portion of the source code is copyrighted material by Cpl\_Filth, used with his explicit permission),
- extracting / rebuilding CC3's ZFX files.

**Credits:** I have to thank Pekka Saastamoinen aka "Cpl\_Filth" for the source code of his CCFx.exe and much more and his advice during the development of this tool. Without his help, this project would have failed completely. And I have to thank Gerry Shaw aka "Tin Tin", Shobun and Cpl\_Filth again for the file format informations. Thanks to Nembo for bug reports on version v2.5.

This tool is published for MacOS and PCs as well. The resulting files created by the different executables of this tool are identical.

### **Known problems:**

- there are some graphical bugs inside the original "Terrain" files: in the CC2 file the sprite #61 has 8 bytes too much, CC3's Terrain file contains 206 bytes too much, and I suppose that the CC4/CC5/CCM/RtB "Terrain" also has a minor graphical bug inside. If you extract the contents of the CC4/CC5/CCM/RtB "Terrain" file and rebuild it with my tool, you will get a 10 bytes larger file, but it will work correctly with the original program.
- CC4 and newer versions do not use any more the Mac-like "Motorola" byte order "Big Endian" but the PC-like "Intel" byte order "Little Endian" for the "Terrain" file. This will lead in trouble if you use a CC2/CC3 "Terrain" file footer-datas directly in newer versions. My tool cannot convert between these two byte orders! It will rebuild the file always in the byte order of the source file.
- My tool will not control the logical integrity of the datas during rebuilding.

- My tool cannot add sprite animation sequences to the footer-data. Use a hex-editor instead. Doing this will create a complete new version of sprite/terrain file which the existing programs and my tool cannot handle.
- This tool cannot analyze nor extract CC1-sprite files.
- CC4 introduces \*.spr files to store vehicle shadows. In some cases the "color-depth ?" value is varying in these files between 63 and 64. During repacking this tool will always set this value to 63.

### Definitions:

- graphical artwork of the CC series is stored in files of different formats.
- These formats vary between the different CC versions, but from CC4 on and newer most of the file formats have not changed any more.
- Some of these files contain only one single graphic, most of these files are called "texture files" (for example some map graphics).
- The other part are graphical archives storing many graphics, sometimes together with a directory list or other informations.
- CC1 (!) uses for storing multiple graphics in a single file the "sprite file" format, indicated by the first four bytes "SPRI". CC1 stores the pixel datas in a different format than all other CC games. CC2 uses 16-bit for each pixel.
- The CC2 file format survived during the entire series until now (in CC4 and newer the byte order changed and each sprite pixel data entry is terminated by an additional trailing zero byte).
- A "16-bit sprite file" contains always 5 parts, I will refer to these parts as "**Sections**":
  - o Header (8 bytes)
    - Header-ID (4 bytes): "SPRI" or "IRPS"
    - Version-ID (4 bytes): always 00000001hex in CC2 and Terrain-files. \*.spr files can have here a 11 different values: 00000001h, AE7800h, AE7A00h, AE8400h, AE9E00h, EE7A00h (airplanes in CC4/CC5), FE9700h, FEA600h, FEBD00h, 12E7800h, 12E7A00h. Meaning of these values is still unknown.
  - o Directory (10 bytes)
    - Directory-ID (2 bytes): 03E8hex = 1000 (decimal)
    - Number of sprite graphics in the "Sprite section" (2 bytes)
    - Number of animation sequences in the "Static animation section" (2 bytes)
    - Number of animation sequences in the "Direction-oriented animation section" (2 bytes)
    - 2 bytes of unknown purpose. Cpl\_Filth supposes that it might be a color depth definition, but it might be also the maximum length of direction-based animation sequences.
  - o Sprite section (varying)
    - SpriteSection-ID (2 bytes): 03E9hex = 1001 (decimal)
    - Sprite entries
  - o Static animation section (varying)
    - StaticAnimationSection-ID (2 bytes): 03EAhex = 1002 (decimal)
    - Sprite sequence entries
  - o Direction-oriented animation section (varying)
    - DirectionOrientedAnimationSection-ID (2 bytes): 03EBhex = 1003 (decimal)
    - Sprite sequence entries
- The format of the "sprite entries" differs between CC1, CC2/CC3 and CC4/CC5/CCM/RtB. But the principle is always the same: each sprite entry contains of three parts: the header

(describing the image size and hotspots), the line offset table and the pixel datas in 16-bit color (since CC2. CC1 stores them in some kind of 8-bit).

- The format of the "sprite sequence entries" is common to all versions.
- "Static animation" sequences have the format:
  - o number of entries in sequence list (2 bytes)
  - o style indicator (of unknown purpose to me) (2 bytes). In case of the \*.spr files these bytes are always filled with FFF2h.
  - o 4 zero bytes of unknown purpose. In case of the \*.spr files these bytes are always filled with F51Ch and 0068h.
  - o sequence list (2 bytes for each sprite entry) of varying length; the sprite numbers are counted from 0.
- "Direction-oriented animation" sequences have the format:
  - o number of entries in sequence list (2 bytes)
  - o 4 zero bytes of unknown purpose. In case of the \*.spr files these bytes are always filled with FFF2h and F51Ch.
  - o sequence list (2 bytes for each sprite entry) of varying length; the sprite numbers are counted from 0.
- Cpl\_Filth referred to the datas behind the "sprite section" as **"the footer"**. His PC-tool "SprTool.exe" generates always the original "footer" when rebuilding a sprite file.
- My tool extracts the "footer" (that means the "Static animation section" and the "Direction-oriented animation section") in a separate file named "Appendix.bin" without checking it for logical integrity. You can edit this file "Appendix.bin" with my tool and it will be integrated in the sprite file during rebuilding. Sorry to say that "Appendix.bin" files created from \*.spr files can not be edited due to a compiler-related list-box limitation. But CC4-RtB are expecting always the same "footer" in \*.spr files, no matter if the vehicle has a turret or not (or is an airplane). I think editing is in this case not necessary.
- For the \*.spr files with their different version-Ids my tool will create a file "VersionID.bin" containing 4 bytes (the version-ID), which will be read when the sprite file is rebuilt. If this file is not present, the tool will use the value "1" as version-ID.
- Any program of the CC series will abort if it encounters a sprite file containing a "footer" with a different number of animation sequences, but you can add freely sprites to the "sprite section" and sprite numbers to the sequence lists in the "footer".
- CC programs do not refer to all sprites in the "Terrain" file via an animation sequence in the "footer", for example the crosshairs in this file have no corresponding entry in any animation sequence.
- Sorry to say that only CC2 displays animated objects out of the "sprite files". Look at the VL-flags of CC3 and newer: despite they are defined as a 20 entries long animation sequence in the file "Terrain", they are not animated during play at all, only the first sprite defined in the animation sequence is displayed.
- But in CC2 you can have animated VL-flags and animated trees. The fact that the trees can be animated is new to us.

### Using the program:

The main window of this tool is divided into four parts using the tab-panel technique: one part contains buttons for analyzing files, the next part the extracting of "16-bit sprite files", the third part the rebuilding of "16-bit sprite files" and the last part the extracting/rebuilding of "32-bit sprite files" (TerrainA.stm files). The program can determine the byte order of the "sprite file" and will handle it correctly.

Extracting datas out of a "16-bit sprite file" will store the graphical contents in uncompressed 16-bit TARGA graphic files in a separate folder. The names of the resulting files will be "Image0001.X.Y.tga" where X stands for the hotpoint in horizontal direction and Y for the hotpoint in vertical direction. The hotpoint indicates the pixel of the sprite graphic which is used as the "center" when displaying it during game play. I have added radio buttons for selecting to have the image filenames counted from 0 (compatible to the internal logic of the game) or from 1 (compatible to Cpl\_Filth's "SprTool.exe").

The tool will store all datas of the "footer" unchanged in a separate file "Appendix.bin". An additional "footer report" file will be produced, compatible to MS-Excel (plain ASCII, TAB-separated columns, CR-terminated lines). This "footer report" file is only for your convenience when checking the animation sequences. It is not used during rebuilding.

There is a separate button present which will bring a window to front where you can select the special colors used during extraction. The sprite entries contain not only pixel datas but also special definitions for transparency areas, shadow areas, color filling areas and a 2nd shadow area (and more special level definitions in the files Soldier/SoldierB). As suggested by Cpl\_Filth and used by his tool "SprTool.exe" the transparency area color is set to white and the shadow color is set to a special light green. The purpose of the color filling area is: it is used to indicate that CC must not produce a color when displaying the sprite. Example: the crosshairs: the area between the two black circles is filled with a varying color during play depending upon the "reachability" of the selected target.

And you can edit the "Appendix.bin" file. When saved, an automatic backup of the previous version of the file will be done. **Remember: the values entered in the sprite sequences are always counted from 0.** If you enter values of sprites which are not defined in the sprite table the CC game will crash at runtime.

For Mac-users I have integrated the extraction of datas out of \*.ZFX archives (CC3) and \*.AZP archives (CC4 and newer) and the extraction of graphics out of the resulting \*.fx files (CC3 and newer) and \*.tex files (CC4 and newer). PC-users please use the already existing tools available on the internet. CC3's \*.ZFX archives can not only be extracted, but also rebuilt. These \*.ZFX files have a fixed size header (number of files, filename table with 600 entries, offset table with 600 entries, 2400 bytes of unknown purpose) followed by the file datas. My tool will export the 2400 header-bytes of unknown purpose into the file "2400BYTE.BIN". This file must be present during rebuilding and will be not imported as a file. My tool will only import readable and visible files if their filename does not start with a point ".". For full extract/repack of \*.AZP and \*.TEX files please use my tool RtBTool v2.1.

Rebuilding of the sprite files can be done in the third tab-panel of the main window. You must select a folder containing the datas. All TARGA graphics must be in uncompressed 16-bit. The tool will take them in the sequence as they are numbered from "Image0000.X.Y.tga" onward. X stands for the hotpoint in horizontal direction and Y for the hotpoint in vertical direction. The last file to be added will be the "Appendix.bin" file. Valid filenames for import must start with "Image" and must end with ".tga". All other files in the selected folder will be ignored by the tool. The tool can skip gaps in the filename numbering. Since v2.0 the tool will handle the TARGA-header informations correctly. But if you want to have the same behavior as Cpl\_Filth's tools you can determine that the TARGA-header informations will be ignored. Pixel data transfer will be then done in the sequence as they are stored in the TARGA data area.

If the file "Appendix.bin" is missing, the tool can only add a "footer" from scratch for the CC2/CC3 vehicle shadow files ("VehB####", "VehS####", "\*.nsd" and "\*.zsd" files). Turretless CC2 vehicles requires 32 sprites (for 32 shadow directions) for correct "sprite file" generating. Turreted CC2 vehicles have 32 gun/turret shadows in addition, so in this case the total number of required sprite graphics is 64. CC3 vehicles require always 64 sprites.

When rebuilding the tool will always generate the header in the correct format for the given "sprite file" and will adjust the number of sprite graphics according to the number of TARGA files in the folder. So you can expand the number of sprite graphics in the "Sprite section" and you must not (but you can) overwrite existing sprites. The tool will set always the number of animation sequences to the correct values of the given CC version ignoring the real numbers in the file "Appendix.bin". If you want to setup a complete new version of "Terrain" file for CC9 or so (with more animation sequences) you must use a hex-editor to correct the header of the produced output file. Such "expanded" files will not be recognized correctly by existing CC programs (perhaps ?).

When rebuilding CC4/CC5/CCM/RtB vehicle shadow files (\*.spr" files), it is possible to have different version-IDs. If this version-ID is not equal to 1, then my tool will write the version-ID to an additional file "VersionID.bin". This file will contains always 4 bytes (Little Endian encoded). The presence of such a file will be detected during rebuilding, importing these 4 bytes as version-ID for the new created "\*.spr" file.

### **STM file format (CC4-CC5-CCM-RtB) – the TerrainA.stm files**

This file format is used by all the TerrainA.stm files of the CC versions since CC4. Whole file is encoded in Little Endian (PC/Intel like). All "sprites" are single graphics encoded in 32-bit (same color encoding as it is used in the \*.fx files since CC3).

```
// header
ALPH // 4 bytes ASCII containing the string "ALPH" = sprites with alpha-channel
long // 4 bytes, containing value "2"
long // 4 bytes, number of sprites in this file
```

```
// directory with sprite definition entries
// each entry 20 bytes of the format:
long // 4 bytes, HotSpot X
long // 4 bytes, HotSpot Y
long // 4 bytes, image width
long // 4 bytes, image height
long // 4 bytes, offset of pixel datas (from top of file)
```

```
// sprite graphics data
// 4 bytes per pixel: blue, green, red, alpha
// line orientation is top-to-bottom, pixel orientation is left-to-right
```

```
We must use Cpl_Filths algorithm of making 32-bit TARGAS
after reading the CC values myBlue, myGreen, myRed, myAlpha:
    if ( ( 8 * myBlue - 1 ) > 0 ) then
        myBlue = (8 * myBlue) -1
    end
    if ( ( 8 * myGreen - 1 ) > 0 ) then
        myGreen = (8 * myGreen) -1
    end
end
```

```

if ( ( 8 * myRed - 1 ) > 0 ) then
    myRed = ( 8 * myRed ) -1
end
if ( myAlpha <> 0 ) then
    myAlpha = ( 8 * myAlpha ) -1
end

```

To write them to a valid 32-bit color-depth TARGA file, the sequence must be changed:

```

OutFile.WriteByte(myRed)
OutFile.WriteByte(myGreen)
OutFile.WriteByte(myBlue)
OutFile.WriteByte(myAlpha)

```

The TerrainA.stm file of CC4 contains 74 sprites, all newer TerrainA.stm files contains 80 sprites. The "Analyze" part of the main window of my tool can detect which format is used by the sprite file (16-bit or 32-bit) by checking the first four bytes of the file. Extracting/Repacking is only possible for my tool when using 32-bit TARGA graphics (8-bit for alpha channel) (like Cpl\_Filth's tool "Stm.exe", thanks again Cpl! Without his tool I would be never able to know for what I have to look inside the ALPH). Manipulating 32-bit TARGA files with alpha channel can be done using Photoshop.

#### **What is new in v1.1:**

- error corrected preventing the tool from analyzing CC4/CC5/RtB Soldier/SoldierB files,
- radio buttons added to select the numbering of filenames when extracting sprite graphics,
- the tool can now skip over gaps in the input file numbering,
- editing of unused columns in the "Appendix.bin" is now possible, but will have (perhaps) no effect during gameplay.

#### **What is new in v1.2:**

- debugging (error while saving Appendix.bin file after editing).

#### **What is new in v1.3:**

- CCM-compatibility added (CCM was released to the public in August 2004).

#### **What is new in v2.0:**

- TerrainA.stm extraction/rebuilding added. TARGA-header interpreting corrected.

#### **What is new in v2.5:**

- \*.spr extraction/rebuilding added. \*.tex file extraction now compatible to my RtBTool.

#### **What is new in v2.7:**

- the tool will now respect the sprite entries' byte-size definition when extracting to TGA. This will avoid trouble with the bug-filled original CC3 Terrain file. Pixels exceeding the sprite's width or height will be ignored. A report after extraction will be shown in a dialog box (will only show up when extracting original CC2 / CC3 Terrain file).

#### **What is new in v2.8.1:**

- \*.nsd / \*.zsd extraction/rebuilding added.
- \*.zfx extraction/rebuilding added.

**This tool is still under development. All in all you are using this tool at your own risk. The author is not responsible for any damage or loss of data this tool will cause.**

Mafi

closecombat2@claranet.de

<http://www.geocities.com/cc2revival/>

<http://www.closecombat2.claranet.de>

<http://members.fortunecity.de/closecombat2/>