Author: Mafi – closecombat2@claranet.de
Title: CC2Tools – The Manual
Actual version: v5.02
Last update: July 2009 (CC:WaR/TLD BTD-file support for shrinking/converting)

Close Combat 2 "A Bridge Too Far"

CC2Tools - The Manual

(Mac- & PC-version of CC2)

What it is

"Close Combat - A Bridge Too Far" (abreviated 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 (later by UbiSoft, then by Destineer) for PCs only (up to day CC3, CC4, CC5, CCM, RoadToBaghdad, CCRAFRegt, CoI, CCMT, CC:WaR, CC:TLD). The game was released in 1997 on a hybrid-CD, running on PCs and under the MacOS 7.5 up to 9.2.2 / MacOS X 10.2.6 up to 10.4.11 (in Classic environment on PPC-Macs) as well. Later (localized) releases of CC2 were for PCs only. A trial demo of CC2 was also released in 1997 (Mac & PC).

Credits and all my thanks to the following programing gods:
Mick Conmy (mick xe5) (for his excellent texts on his site http://users.intrepid.net/~mcconmy/), Adam 'The Man' D'Arcy (who made public the file formats for the CC2 map graphics), Gerry Shaw (aka TinTin) (for his great tools Texture Maker v.2 and TM3), Vincent Viaud (for solving the LOS file structure), Andrew (The Naked Foot) Glenn (for his great MapMaking Guide), Escobar for his GadgetX/I- and IntrfaceX/I-tools, The Other Dave (David R. Tidy, for his CC2Faq.wri), NL_Oxcart, George Thanos, Taki (for his PacificFront mod, my first encounter to CC2-editing), GS_Marcks, Chris Ellens (for his great CCEdit map-making-tool for MacOS), Cpl Filth (for his great tools and a lot of help), Frantz 'Fritz' Pergolini, Robert Valerian 'Cappy' Ellison Ralph (for his Gadget-catalogue), Marcus 'Zorbo' Hofbauer, Piotr 'Czolg' Lewandoski, Kyle Scott 'Fish', Riccardo Mariani, Hikehara, Mizuchi, Kelly Kranendonk, Konrad, David Vilmen, IChrist (for his CC5-CC3 elements table), Han Bos (aka NL_Attila), Boar ... and all the people not mentioned above, who helped CC2-editing to come true.

Special thanks this time to Ken Scott, who made the first steps in unveiling the CC2-number-of-teams-limit and to Francisco Arias 'Nembo', ANZAC_Tack, tejszd and all the other beta-testers!

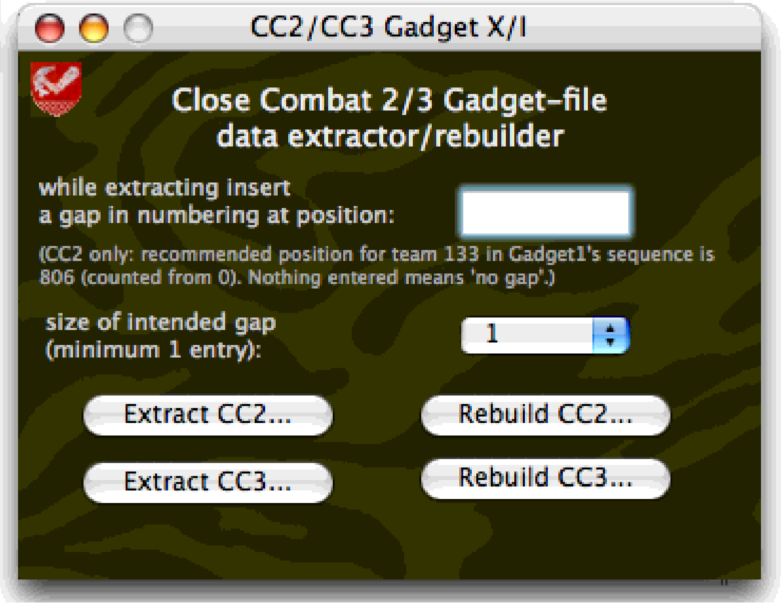
Purpose of the program

The goal is to convert existing RtB/CC5, CC4, CC3 and CC2 maps to other CC-version’s format to use them on the Mac or for other purposes and to have better MacOS-like access to

other CC2 files not already covered by Chris Ellens' tool "CCEdit" or my CC-map editor "5CC". Since 2006 it is also used for stitching smaller maps together get very big maps.
CC2Tools can do for you:

- convert all map graphic files of all CC-versions (from CC2 to CC5/RtB) into TARGA,
- convert from TARGA into CC2-BGMap, CC2-MMMap, CC2-OVMap, CC2-Vehicle-Textures, CC2-Roof, CC3/RtB-bgm, CC3/RtB-mmm, CC3/RtB-ovm.
- tiny CC2-bridge-file creating (import and conversion of 1 or 2 TARGA-graphics),
- shrink all CC2-CC5/RtB map files in their size,
- and the main part is: shrinking and converting from RtB/CC5/CC3 into CC2 format (and back) in ONE step,
- and making the terrain-elements conversion from CC5 to CC2 or CC3 to CC2 or in any other way if you define an external translation table (TAB-seperated text file, MS-Excel-like).
- translation tables included for CC3 to CC2, CC5 to CC2, RtB to CC2, CC2 to RtB, CC5 to RtB, RtB to CC5,
- expanding a CC2/CC3/CC4/CC5/RtB map's size without changing the file format (for map's data/txt-, background/bgm- and roof/rfm-files),
- rotating map's data, background, roof and bridge files 90° counterclockwise and flipping them vertically,
- joining two maps' data, background or roof files to one file,
- creating plain map data files from scratch,
- and it can do some manipulations to increase the number of teams in CC2,
- editing the CC2-'stri'-resource entries (MacOS version only),
- and can export / rebuild the CC2 / CC3 files Gadget0, Gadget1 and Intrface.cc2 as well as the CC3-CC5/RtB CCImages.fx / CCImages.pix export and rebuild,
- delete / add the end-of-line delimiter char "LF" to one single text file or to the entire CC2 installation on your HD.

Gadgets ex- and importing function


CC2/CC3 Gadget X/I

Close Combat 2/3 Gadget-file
data extractor/rebuilder

while extracting insert
a gap in numbering at position:

(CC2 only: recommended position for team 133 in Gadget1's sequence is 806 (counted from 0). Nothing entered means 'no gap'.)

size of intended gap
(minimum 1 entry): 1

Use this window to split off the little graphical "gadgets" out of the files "Gadget0" and "Gadget1" (since program v4.0 for both CC2 and CC3, the two versions only differs in the internal byte order (CC2: Big Endian encoded, CC3: Little Endian encoded)). Pressing the

button "Extract..." will show up an input file selection dialog box. You can determine a "gap position" when the resulting files are exported. If you enter nothing in the editfield, no gap will be inserted in the numbering of the resulting files. Please use different folders for output for each "Gadget#" file you want to split off. Then you will be able to rebuild these files from the resulting output by pressing the button "Rebuild...".
Output of the "Extract..."-button will be

- when splitting "Gadget0": 700 graphics files (16-bit uncompressed TARGA). Resulting file names will be of the form "gadget0-004-0007.tga", where the last four digits stands for the sequence inside the file "Gadget0" and the hex-digit between the two dashes shows the visual width of the graphical content of this gadget.
- When splitting "Gadget1" the result will be 939 files of the same format as above.

If you don’t change the name of these resulting files, the rebuilding of the "Gadget#" files will work properly using the button "Rebuild...". It is not recommended to change the size of the gadgets.[1] But you can have more gadgets inside the "Gadget#" files. This cannot be done using Escobar’s tools. So I was forced to implement the "gap functionality":

If you want to have more than 133 teams in CC2, you can simply expand the list of teams inside the base files "Teams" and "TeamDesc" (using a text editor or Ms-Excel, these files are TAB-seperated text files, CR-delimited (MacOS-like)). But the program CC2 will crash during startup because it is looking for additional gadget entries in the files "Gadget0" and "Gadget1".

To have an additional team, it is sufficient to add a gadget to the list of gadgets in the approbiate folder for the rebuilding of the file "Gadget0". My tool will detect these additions and place them at the end of the file "Gadget0".

In case of the file "Gadget1" you will also need an additional gadget at the end of the file and (different from "Gadget0") you will need an additional gadget in the middle of the files’ sequence at position 806 (if you count them from 0). The reason is that in the file "Gadget1" there is a large and a small gadget for each team.

So you need a gap in the numbering of the files when splitting off "Gadget1". To avoid manual filename renaming I added the "gap position" functionality.

After expanding the files "Gadget0" and "Gadget1" in this way CC2 will work with more than 133 teams. Not tested yet how many teams are possible. Maybe there is a memory overflow limit.

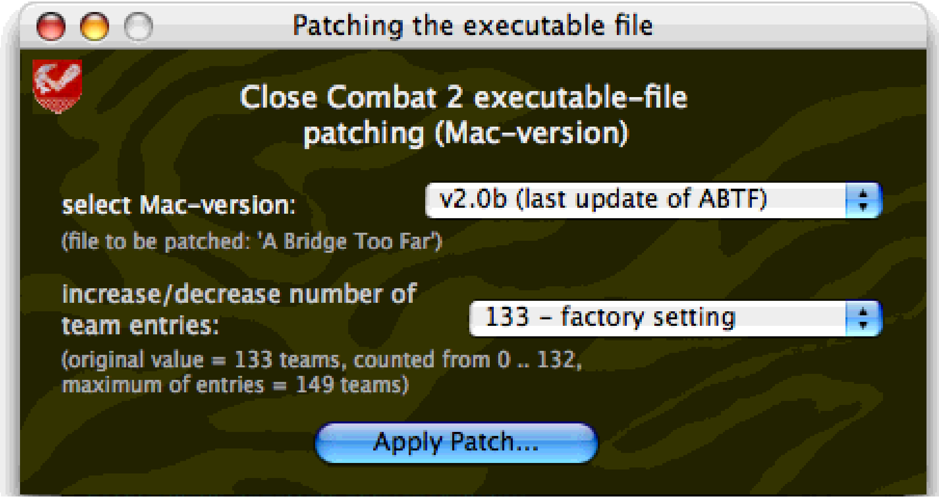
As Ken Scott discovered, you can shrink the number of teams down to 2, but you must not change the files "Gadget0" or "Gadget1". The reason why is: the offset of the gadget of team 0 is coded in the object code of CC2 (Mac and PC). So we are coming to the next task:

---

[1] But if you do so, don’t forget to change the visible width entry (in hexadecimal) in the filename.

Patching the executable application

We need to change the number of teams inside the object code of CC2 (Mac and PC version) if we want to have more than 133 teams. See Gadget-description in chapter above. As discovered by me, the offset of the second team gadget list of file "Gadget1" is hard coded inside the object code. The number of the gadget is 806 (if you count from 0, see Cappy-R's catalogue). The number inside the object code must be changed to 807 if you want to have an additional team gadget. The position of this number differs between the CC2 versions:



Patching the executable file

Close Combat 2 executable-file patching (Mac-version)

select Mac-version: v2.0b (last update of ABTF)

(file to be patched: 'A Bridge Too Far')

increase/decrease number of team entries: 133 – factory setting

(original value = 133 teams, counted from 0 .. 132, maximum of entries = 149 teams)

Both (Mac & PC) versions of CC2 store the informations about the teams in the base file "TEAMS". The number of teams is limited there only by the number of entries in this file. As Ken Scott discovered, you can shrink the number of teams down to only 2 teams, and CC2 will still work. Expanding makes it necessary to expand the number of entries in the base file "TEAMDESC", too. And you MUST expand the number of gadget images for the additional teams in the graphics file "GADGET0" by adding at the end. Same to do with the file "GADGET1". And you MUST insert in the file "GADGET1" smaller gadgets for this teams in the middle of the file at entry 806! This is the starting point of the larger gadgets of "GADGET1". In the executable (ABTF on Mac / CC2.exe on PC) you must patch this starting point 806 in the object code. The ending point of reading the smaller gadgets takes CC2 from the number of teams in the file TEAMS. The offsets in the object code where you have to patch is:

Version	Mac Value = 0326hex = 806	PC Value = 2603hex = 806
Demo	Offset = 03F8F6hex	Offset = 0602B7hex
First version after install from original CD-ROM	Offset = 03FD16hex	Offset = 0609BAhex
Last available update	Offset = 03FD02hex	Offset = 060967hex

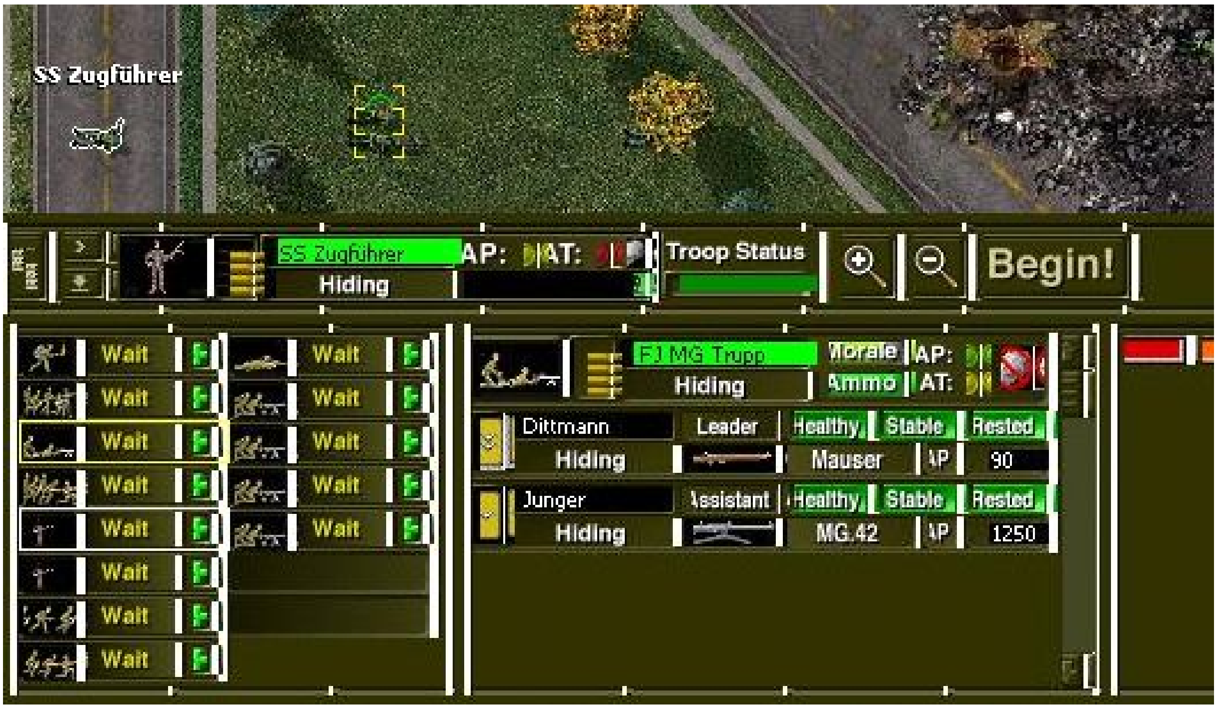
My tool will do the patching, but you must select the version of your CC2 properly. And you can select how many teams you want to add.
If you want to have more vehicle teams, you will encounter that there is no patching possibility (up to now). All additional vehicle teams will use the graphics of the first vehicle team (Sherman). Looks like a wrap-around / modulo function in CC2.

Problems when expanding the "Teams" number

Known side-effects when expanding the number of teams are:

- you might loose the CC2-battlemaker functionality of the game (PC version only),
- you might encounter strange sprite artefacts on th eleft side of the screen when scrolling to the right side,
- not all of your implemented teams are available for user selection during play.

The following example screenshot of an error when expanding the number of teams in the files "Teams" and "TeamDesc" reached me in January 2005 (PC version of CC2.exe):



Picture: error screenshot by Nembo.

The error occured because the non-patched original CC2.exe was used with expanded base files and expanded "Gadget1". The file "Gadget0" was not expanded. So here an additional briefing how to implement additional teams to CC2-ABTF, as discussed with the adding of one single team:

1. Decide how many teams you wish to add to the existing ones (remember: maximum is 149 teams in CC2); in our case: 1 team, to have team class 133.
2. Expand the base files "Teams" and "TeamDesc".
3. Look what CC2.exe / ABTF executable version you have. I think you have the CC2.exe from the original CD updated to the lastest version v2.0b using the patch from the internet.
4. Use this tool to patch the executable program: select the CC2.exe version you have, and then: select in the popup-menu the number of teams you want to have. Then press the button "apply patch" and give the new *.exe file an independent name (for exmaple "CC2-134.exe").
5. Extract the "GADGET0" file using this tool.
6. Extract the "GADGET1" file using this tool. Watchout! For every team to be added there must be room inside the gadget numbering behind index 805 (counted from 0). In the Gadget X/I-screen please enter the number of the starting position for this

gap. And the size (number of entries in this gap) have to be entered: in our case enter "806" as starting position, and size of gap"1" (one gadget to be added).

When extracting then the "Gadget1"'s images will be extracted with the original gaget #806 getting the new index number #807, #807 getting #808 ... and the last #938 getting #939!

It might occur that you want to have an already "expanded" "Gagdget1" file to be expanded further. In this case the strating position must be changed.

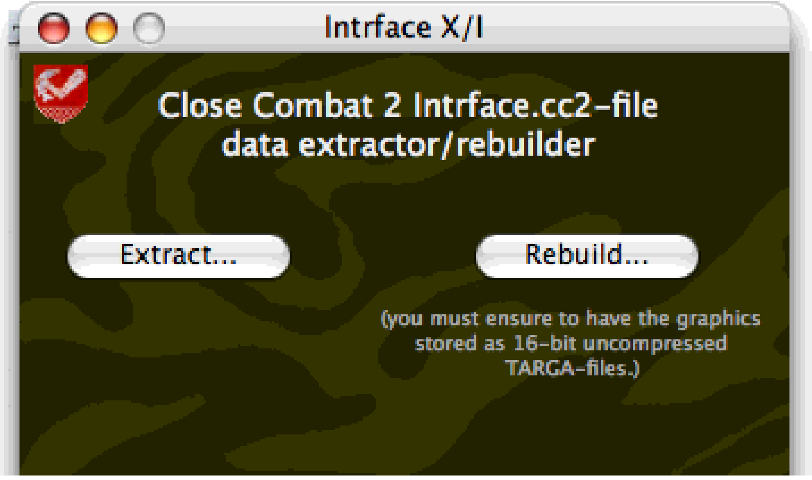
Example: you have finished it with your team 133 and now team 134 is to be inserted. Then you must set starting position "807" and gap size "1". I know its not simple and easy to imagine if you have already coded it as a program. I hope you get through it.

7. Adding new graphics to "GADGET0": its easy, just adding a new TARGA at the end of the list. This image must have the name Gadget0-hhh-0700.tga (dont forget the .tga), where hhh stands for the hex-encoded "visible width of the image" (not identical with image width), in our case it should always be "029": so the correct name is Gadget0-029-0700.tga .
8. Adding two new graphics to "GADGET1": you must insert one at the end of the "029" list and one smaller at the end of the "01C" list (this is the end of the whole list at all). Please notify: the TARGAs' index numbering are the last four digits in the filename!!! As you did correctly the inserting point behind the original "029" list is 806: filename must be Gadget1-029-0806.tga. But for the image to be added at the end of all list, the image's index must not increased by 1, but by 2, because we have already inserted an image: last original gadget is Gadget1-01C-938.tga. So the new one must be Gadget1-01C-940.tga !!!! And that is the error! You have entered Gadget1-01C-939.tga, this will overwrite the existing Gadget1-01C-939.tga. And one small gadget is missing at the end of the list, crashing the program after loading the map and organizing the user interface (if you have the additional team aboard).
9. Repacking both gadget files with this tool. Ready to use.

Why are there the small white strips at the right side of each gadget? All graphics in the "Gadget#" files are expanded to a width being a multiple of 8. The unused space in these images is left white. That is the reason why the "Gadget#" files contain in their directory not only informations about every gadget’s width and height (and image data offset), but also the **"visible width of the image".** If this entry is larger than the actual painted width of the gadget, the white area will become visible, if you have the PC version of the game running. The Mac-version of CC2 will not show the white strips and will interprete the white area as "transparency". The both versions of the game differs in treating the gadget painting.

Intrface ex- and importing

Same as with the "Gadget#" files, you can extract the graphical content of the file "Intrface.cc2" and rebuild it unless you ensure: you will always need 19 graphics, each of 16-bit uncompressed TARGA.


Intrface X/I
Close Combat 2 Intrface.cc2-file data extractor/rebuilder
Extract...
Rebuild...
(you must ensure to have the graphics stored as 16-bit uncompressed TARGA-files.)

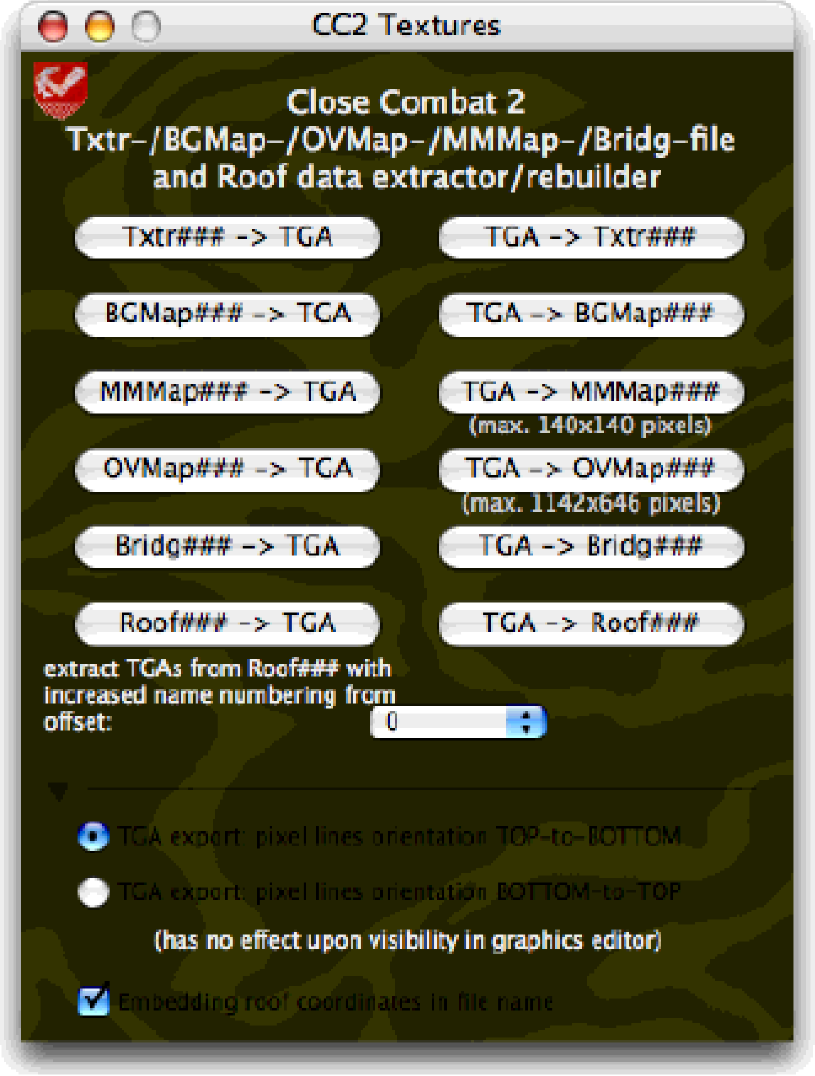
The file "Intrface.cc2" is necessary for the PC version of CC2, and most of the existing CC2-mods have only this file. The MacOS-corresponding file is the file "UI". This file has a resource fork, which cannot be viewed by PC users (and therefore cannot be rebuilded there). So what to do: if you have a "Intrface.cc2" and want to change it into an "UI": export the "Intrface.cc2"-graphics, use a graphical program on the Mac, paste the graphic into the clipboard and use Apple’s ResEdit 2.1.3 to insert the graphics into the resource fork’s "PICT" resources. Last graphic must be placed into the "ppat" resource. So you can convert any custom "Intrface.cc2"-file into a Mac-"UI"-file. For more informations see my updated guides "CC2Guide-MacResourcesPat-v3.pdf" and "CC2Guide-UI_Intrface_OvData.pdf" from my site http://cc2revival.npage.de/ or http://closecombat2.fortunecity.com/

CC2 Texture ex- and importing

Most of the graphics (but not all) are MacOS born files. Their file format is mainly "Big Endian" (exception: "Intrface.cc2"). These graphics use an identical scheme: a 4 byte header ID is followed by a table with data descriptions and then the umcompressed graphical datas (each pixel 16-bit long). Files containing only one graphic are: all Textures (files "Txtr###", header ID "txtf"), "BGMap###" (header ID "MAPI"), "OVMap" and "MMMap" (header ID 4 zero bytes). The roof- and bridge- files (headerID "ROOF" and "BRDG") contain several graphics and therefore they have a longer offset table incorporated. Using the screen will show you the possibilities of my tool. If you convert back to CC2 file format you must ensure to have again 16-bit umcompressed TARGA file format for your graphics, otherwise the program will fail. Something to notify: since v4.0 my tool will interprete the TARGA header correct, respecting the pixel line orientation (top-to-bottom or bottom-to-top) as it is encoded in the TARGA header. See original TARGA file format definitions (can be found on the internet[2]). When exporting you can determine in the lower section of the window (after clicking on the disclosure triangle at the lower left edge) which pixel line orientation to use.

---

[2] for example: http://netghost.narod.ru/gff/vendspec/tga/tga.txt



extract TGAs from Roof### with increased name numbering from offset: 0

● TGA export: pixel lines orientation TOP-to-BOTTOM

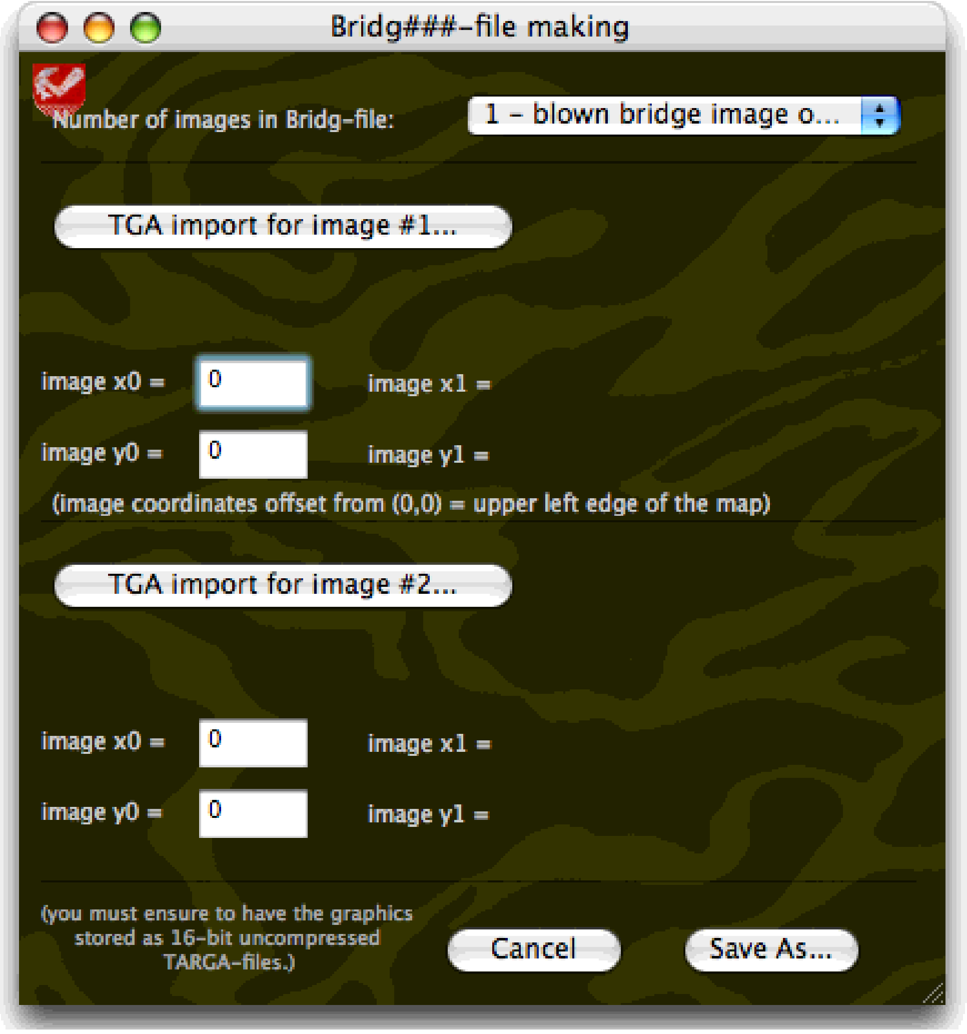
○ TGA export: pixel lines orientation BOTTOM-to-TOP

(has no effect upon visibility in graphics editor)

☑ Embedding roof coordinates in file name

In case of the Roof- and Bridge-files you must select a folder where the resulting output will be stored. For rebuilding Roof-files with my tool the roof coordinates must be exported within the file names. To merge several roof files, store their extraction outputs in one folder. To avoid overwriting already extracted files you can set a different file name numbering offset (default will be 0). Coding of Roof-files can be done with my CC-map editor "5CC" (for Mac and PC available), Chris Ellens' map making tool for MacOS, "CCEdit 1.4a5" (http://www3.sympatico.ca/ellens/ccedit/), or with Cpl_Filth's tool "Groof2.exe" on a PC.

For Bridge-files I have added a seperate window (v3.02): when you press the "TGA -> Bridg###"-button, the following window will appear:



Number of images in Bridg-file: 1 - blown bridge image o...

image x0 = 0
image x1 =
image y0 = 0
image y1 =

(image coordinates offset from (0,0) = upper left edge of the map)

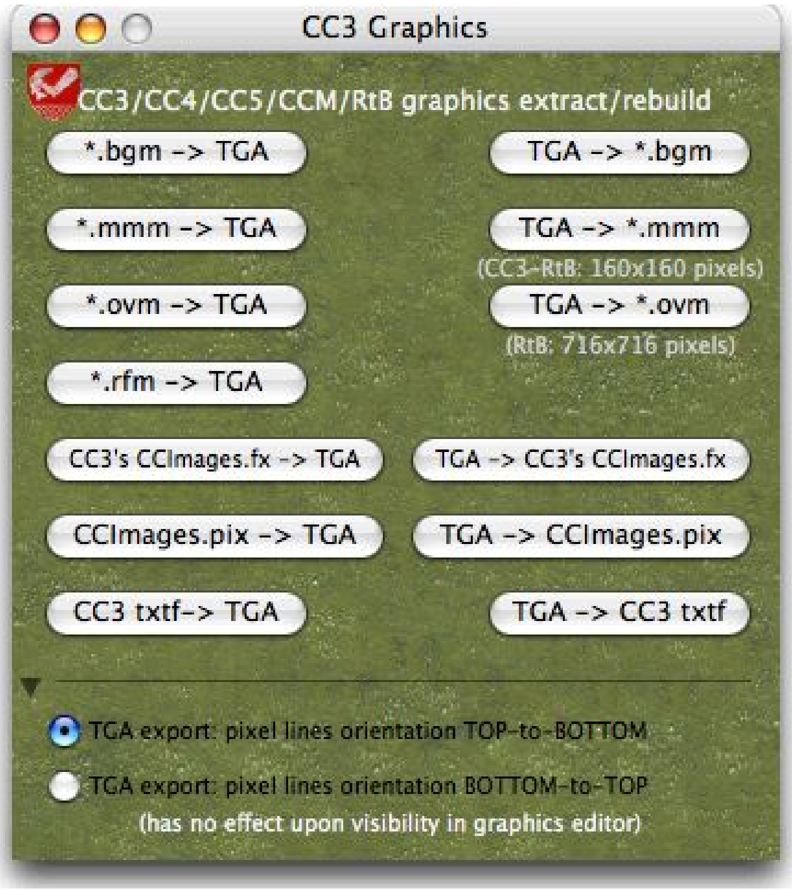
image x0 = 0
image x1 =
image y0 = 0
image y1 =

(you must ensure to have the graphics stored as 16-bit uncompressed TARGA-files.)

First you must select how many images the Bridge-file should contain (1 = blown bridge image only; 2 = blown bridge image and a repaired bridge image). Then you must select the approbiate TARGA-graphic file(s) for input (16-bit color depth, uncompressed). The program will determine and take the size from the selected graphic file(s). The next step is to tell the program the position of this images on your map. Offset is (0,0) = upper left corner of your map. Enter the images' upper-left coordinates in the edit fields. If all is set correctly, press the "Save As..."-button to enter the name of the file to be created and save your new Bridge-file. For further informations concerning the Bridge-files see my updated guide: "CCGuide-Bridg###-files_v6.pdf".

CC3 / CC4 / CC5 / RtB / CCMT / CC:WaR Texture exporting/rebuilding

All of the graphics of the PC-only CloseCombat game series released after CC2 are PC born files. Their file format is always "Little Endian" (Intel style byte format, reverse byte format). If you want to have (custom) maps intended for CC3 (The Russian Front), CC4 (Battle of the Bulge), CC5 (Invasion Normandy), CCM (Close Combat Marine), RtB (Road to Baghdad) or newer to be converted to CC2, you will first take a look at the background graphics using this window:



● TGA export: pixel lines orientation TOP-to-BOTTOM

○ TGA export: pixel lines orientation BOTTOM-to-TOP

(has no effect upon visibility in graphics editor)

Simply the same as above, but remember: the difference between a CC5- or CC3-*.bgm-file to a CC2-BGMap is mainly the byte format of the pixel datas and the byte format of the numbers encoded in the header. The header ID is nearly the same: BGM has "MAPI", MMM and OVM have 4 zero bytes. The header ID of RFM is E002F001hex. The collection of user interface graphics in the CC4/CC5/CCM/RtB-file "CCImages.pix" has the header ID "PICS", the CC3 file "CCImages.fx" has no header ID.

Since v4.10 it is also possible to convert CC3’s "TXTF" format to TGA and back. This file format has survived until CC4/CC5 for flame thrower animation textures. These CC3-TXTF files have a header 8 bytes longer than the CC2-TXTF files (Hotpoint-X and Hotpoint-Y added). CC3 uses this format for the graphics inside "Guns.zfx" and "Wrecks.zfx".

Something you must know: roof graphics in CC2 are always full colored rectangles. In the newer versions CC3 – RtB these roof graphics show only the roof and interior without the surrounding terrain. These terrain region is left white (transparent). Therefore these roof files contain not only an offset table but also a table with vertices, as told to me by Cpl_Filth (thanks Corporal!):

Author: **CplFilth** (---.oulu.fi)
Date: 10-08-2003 18:17

CC3-CC5 Roof-File format

All values decimal and MSB-first(big endian) unless otherwise indicated. All integers are 4 bytes long.

```
Header ( a total of 16 + 132*roofs bytes )

        Once:
                integer E0 02 F0 01 // header ID
                integer number_of_roofs // ie. roofpairs
                integer 0
                integer 0
        for ( each roofpair )
                integer 2 // number of roof images for this entry
                integer number_of_vertices
                for ( i = 0 ; i < number_of_vertices ; i++ )
                        {
                        integer vertex.i.x-coordinate
                        integer vertex.i.y-coordinate
                        }
                for ( i = number of vertices ; i < 12 ; i++ )
                        {
                        // 0xCDCDCDCD indicates this vertex is not used
                        integer 0xCDCDCDCD
                        integer 0xCDCDCDCD
                        }
                integer 2*roof_width
                integer roof_top_left_x-coordinate
                integer roof_top_left_y-coordinate
                integer roof_bottom_right_x-coordinate
                integer roof_bottom_right_y-coordinate
                integer offset_of_exterior_data // from beginning of file
                integer offset_of_interior_data // ditto

Data (the rest of the file) // standard cc format graphical data ie. pretty
much TGA

roof0_exterior_data
roof0_interior_data
roof1_exterior_data
roof1_interior_data
roof2_exterior_data
roof2_interior_data
etc.

That's about it.

CplFilth
```

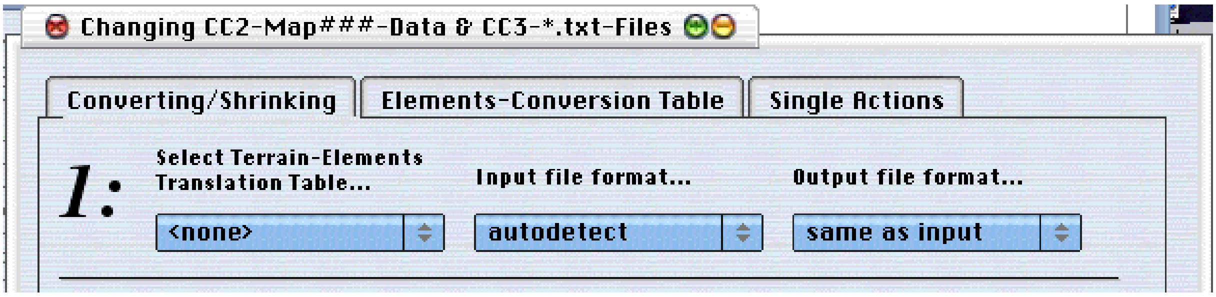
The main task: Map-Converting from CC3/CC4/CC5/RtB to CC2

The differences in the roof file formats is one of the main difficulties in converting a CC3-CC5-map into a CC2 map. For all other things there are tools on PC already available. To do the job there are some other problems to solve, too:

- CC3-CC5/CCM/RtB maps are encoded in a different graphics file format (LITTLE ENDIAN), CC2 graphics are encoded in Big Endian,
- CC3-CC5/CCM/RtB map data files depend on a different terrain-element definition,
- CCM/RtB maps are too large for CC5 (a standard RtB map is 40x40 mega tiles large, CCM's BIG maps are 160x40 mega tiles large),
- CC5 maps are too large for CC3 (CC3 limit is 30x30 mega-tiles), RtB/CC5-CC3 maps are too large for CC2 (CC2 limit is 19x19 mega-tiles).[3]

So I put it all together into one (now nearly overcrowded) window. The input files are always detected by their header if they are from CC2 or not. The steps are as follows:

1. select if you want to have a translation in the terrain element numbers. I have included two translation tables (CC5 to CC2, CC3 to CC2) which can be altered freely. Remember: some of the mods use different terrain element definitions. In CC2 these definitions are in the base file "Elements" (TAB-seperated text, CR-delimited). And you must set the output format of the resulting file(s): same as input, CC2 or CC3-CC5.



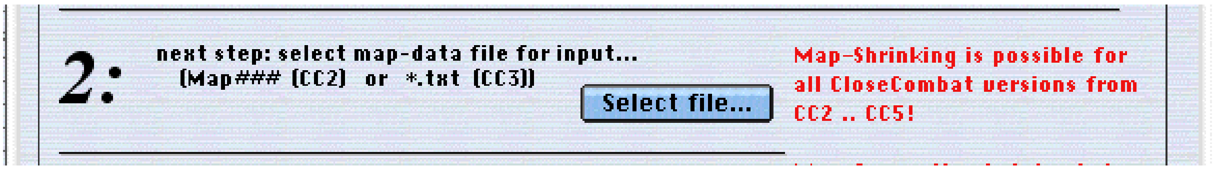
Changing CC2-Map###-Data & CC3-*.txt-Files

Select Terrain-Elements Translation Table... <none>

Input file format... autodetect

Output file format... same as input

2. Then you must select a map data file. These kind of files contains the map's terrain description (numbers, TAB-seperated text file) depending on the terrain definitons of the the base file "Elements" and the elevation description. There are differences in the file format between CC2 and the other versions, and they have different headers (luck!). After this file is imported you can do the next step:

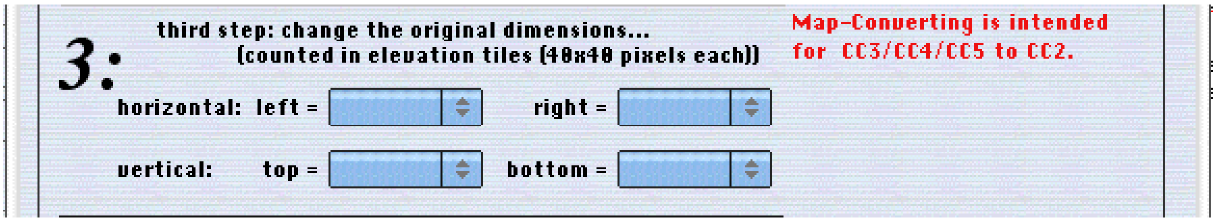


next step: select map-data file for input... (Map### (CC2) or *.txt (CC3))

Map-Shrinking is possible for all CloseCombat versions from CC2 .. CC5!

---

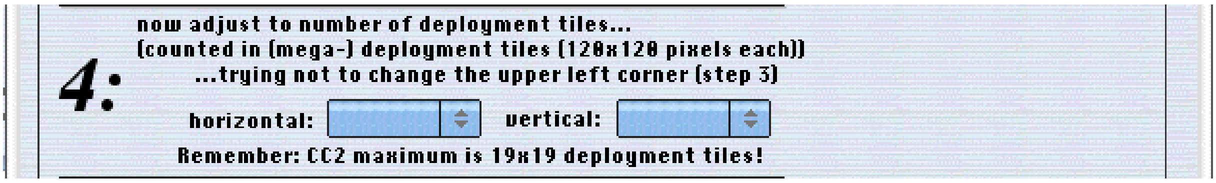
[3] A mega-tile is a graphical subset of a map with a 120x120 pixel dimension. It is also called 'deployment tile' because these tiles are the grid to deploy your troops at start of a battle and the grid for placing the victory locations.



3: third step: change the original dimensions...
(counted in elevation tiles (40x40 pixels each))

Map-Converting is intended for CC3/CC4/CC5 to CC2.

3. you can keep the original dimensions of the map if it will fit to your pupose. In any other case you can select a new upper left offset of the map (popup-menus at the left side) and in step 4:



4: now adjust to number of deployment tiles...
(counted in (mega-) deployment tiles (120x120 pixels each))
...trying not to change the upper left corner (step 3)

Remember: CC2 maximum is 19x19 deployment tiles!

4. you can determine the width and the height of the new cut-out. The width and the height can be entered in step 4 counted in deployment tiles and will be shown up counted in elevation tiles in the popup-menus of step 3 at the right side. My tool will always try to have a width /height in full deployment tiles (that means: a multiple of 120 pixels), but you can set up the upper left edge of your cut-out freely counted in elevation tiles (that means: multiples of 40 pixels).
5. Now the new dimensions are set and you can convert all kind of files to your new dimensions:

- Map-data files (CC2: "Map###" out of the folder "Data/Maps"; CC3-CC5: *.txt files). In case of the CC2 files: I managed to recalculate the bridge-file data addendum at the end of the "Map###" files..
- LOS-files (CC2: "Map###.los" out of the folder "Data/Maps"; CC3-CC5: *.los files),
- Background graphics (CC2: "BGMap###" out of the folder "Graphics/Maps"; CC3-CC5: *.bgm files),
- The minimap and overviewmap files are not shrinked but created new from the original background graphics cut-out when pressing the corresponding buttons. My algorithm used is not so good as the one of Chris Ellens' CCEdit..
- Roof files (CC2: "Roof###" out of the folder "Graphics/Maps"; CC3-CC5: *.rfm files). Remember the roof graphics differences: I solved the white space problem around the CC3-CC5 roof graphics by pasting in the terrain graphics from the original background graphic file (*.bgm). If the program will not find this file, it will ask for it in a seperate dialog. This pasting in will occur always when: converting from CC3-CC5 to CC2 and when some of the roof graphics must be cut-off during shrinking (so I avoided a recalculating of the vertices). Best way to avoid the seperate dialog box: keep all incoming CC3-CC5 map files in one folder. For special handling of roof graphic intersections see v4.03 addendum at the end of this manual!
- CC2-bridge files will also be recalculated correctly during shrinking...
- Like the CC2-scenario files. The only thing I didn't implemented is to take over the original deployment zones. Do this manually. In case the input is CC3 scenario file this file can be converted into (shrinked) CC2 scenario. These CC3 files have the name of their battles (sorry, they have no header). If the input is a CC4 or CC5 map, these scenario files are *.btd files and can be convertedd and shrinked into CC2 files, too (they have a header, thanks to Mick (xe5) Conmy for his advice and description).

Since version v3.09 the converting/shrinking can be done in any direction: CC2 to CC3/CC4/CC5/RtB format and back as well!

The entries in the used translation table from step 1 can be viewed on the second tab-panel entry (and can be changed and saved for further use). On the third tab-panel entry are some single actions for tasks **without** shrinking:

- converting background graphics from CC3-CC5 into CC2 format,
- converting roof files from CC3-CC5 into CC2 format,
- creating CC2 OVMap### and MMMap### files from CC3-CC5 background graphics,
- and the same with resulting CC3-CC5 *.ovm or *.mmm files.

An example: converting an original CC5 map to CC2 on the Mac

I will take Cherbourg Arsenal map. Launch CC2Tools. Open the dialog "Map converting/shrinking", make first tab panel visual. There are 5 steps to do (they are numbered):

1. set translation table to "<select external table>" an search for the file "CC5toCC2.txt" which came with this archive. It will be loaded into memory. Now set ouput format to "Close Combat 2" (THIS is ESSENTIAL).
2. select CC5 map data file for input: "CHRBRGAS.TXT" from original CC5 CDROM; my tool will show then the file format and size of the map.
3. set the new dimension to Left = 4, Top = 34 with a width of 14 mega-tiles and a height of 13 mega-tiles.

4a. press the button "Shrink..." and enter new name for output file "Map400".

4b. press the button "Shrink LOS..." to search for corresponding LOS-file "CHRBRGAS.LOS" and enter name of new LOS-file "Map400.los" (quicker than making a new LOS-file).

5. press button "BGMap..." and search for CC5 background graphics "CHRBRGAS.BGM" from same source as above. Set output file name to "BGMap400".
6. press button "MMMap..." and search again for same CC5 background graphics "CHRBRGAS.BGM" from same source as above. Set output file name to "MMMap400".
7. press button "OVMap..." and search again for same CC5 background graphics "CHRBRGAS.BGM" from same source as above. Set output file name to "OVMap400".
8. press button "Roof..." and search for corresponding CC5 roof graphics "CHRBRGAS.RFM" from same source as above. Set output file name to "Roof400".
9. press button "CC4/CC5-btm -> Scenario..." and search for corresponding CC5 BTM scenario file "CHRBRGAS.BTM" from same source as above. Set output file name to "Scenario".
10. open new "Scenario" with Ms-Excel or SimpleText and change the map name from "Map102" to "Map400" (this must be done, if not, the map will not start),
11. create a folder named "1400", place the new "Scenario" and some suitable existing "AlOOB" and "AxOOB" files in it , too.
12. change your base file "Batnames" and add an entry map 400 as single battle and increase the number of battles (Ugur's method).
13. Place the folder "1400" in the folder "Close Combat:data:data:battles" of your disk image of CC2 original disk, place the file "Map400" in the folder "Close

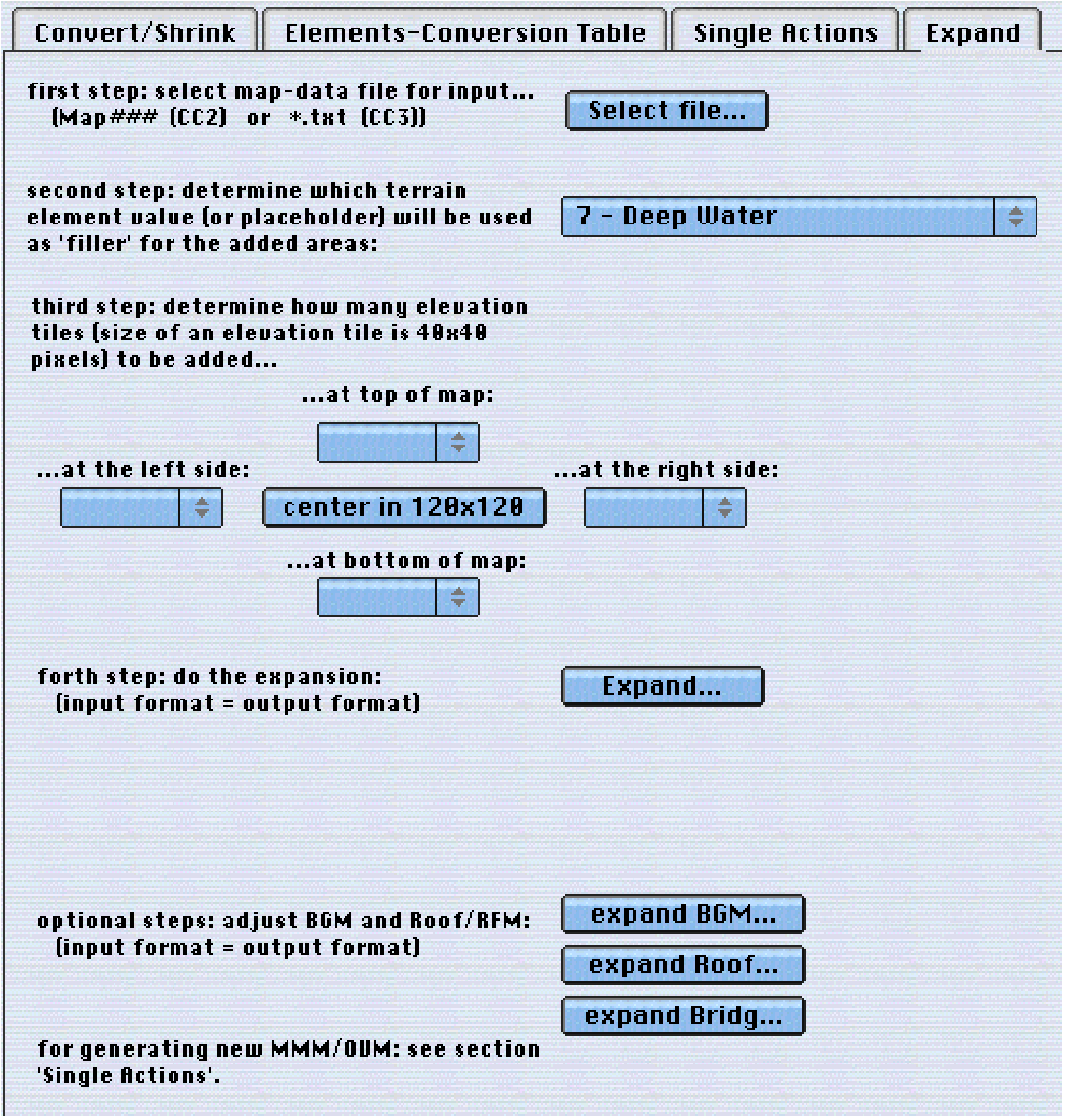
Combat:data:data:maps", place the graphics files "BGMap400", "OVMap400", "MMMap400" and "Roof400" in the folder "Close Combat:data:graphics:maps".

14. Perhaps you will launch CCedit od CClos and generate a different LOS-file, if you need it.

15- Ready, start ABTF and you will have a new map with the lower portion of "Cherbourg Arsenal".

The v3.05 Update

In January 2004 the next game of the CC series was released (in USA only): The Road to Baghdad (a crippled CCM engine). This new game requires extremely large maps (40x40 mega tiles = 120x120 elevation tiles). To use our old CC3/CC5 custom maps, it became necessary (and was suggested by Mick (xe5)) to expand existing maps without changing the file format. I have added a new section to my tool to do the job. In a seperate panel of the "Map converting window" you can (after selecting a map-data-file (CC2: "Map###", CC3-CC5/RtB: "*.txt") determine where to add "black space" at the map’s boundaries: top, bottom, left and right. This adding can be done/is fixed to multiples of 40 pixels, and the popupmenus will count the amount of space to be added in "elevation tiles".



first step: select map-data file for input...
(Map### (CC2) or *.txt (CC3))

second step: determine which terrain element value (or placeholder) will be used as 'filler' for the added areas:

third step: determine how many elevation tiles (size of an elevation tile is 40x40 pixels) to be added...

...at top of map:

...at the left side:

...at the right side:

...at bottom of map:

forth step: do the expansion:
(input format = output format)

optional steps: adjust BGM and Roof/RFM:
(input format = output format)

for generating new MMM/OVM: see section 'Single Actions'.

For Mick’s convinience I have added a "Mick-button" labeled "center 120x120" which will automatically center an input map in a standard RoadToBaghdad 120x120 elevation-tile square.

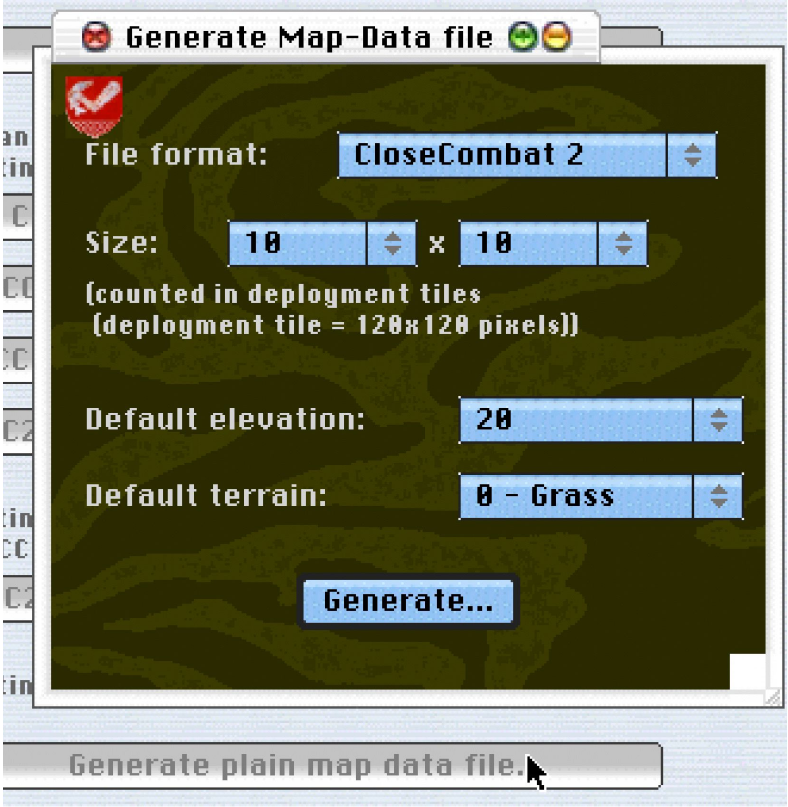
The tool will warn you if you don’t respect the mega-tile size (that means: you must always add multiples of 3 elevation tiles in each direction. Example: adding 1 elevation tile at the left and 1 at the right side will fail, you must at least add 2 elevation tiles at the one side to add 1 elevation tile at the other side.)

The added space around the map will be filled up with a terrain element value which you can determine in a seperate popupmenu. Value 7 (= deep water in all CC games) is the default. But you can also select two other placeholders (value 255 = illegal value in all CC games; plain text "NEW") which must be converted into the value you will need to make the map work.

The lower pushbuttons are intended to correct some graphics files of the map according to the settings you made above: the background graphics file (BGM###/*.bgm) will be filled up with black space, and the coordinates inside the roof- and bridge-files will be adjusted without changing their graphical contents.

For generating approbiate MMM/OVM files for the expanded map please use the pushbuttons in the "Single Actions" panel. And you must use a tool like CCEdit or CCLos.exe to generate a new LOS-file.

The v3.07 Update – generating new map data files from scratch


Generate Map-Data file

File format: CloseCombat 2

Size: 10 x 10

(counted in deployment tiles
(deployment tile = 120x120 pixels))

Default elevation: 20

Default terrain: 0 - Grass

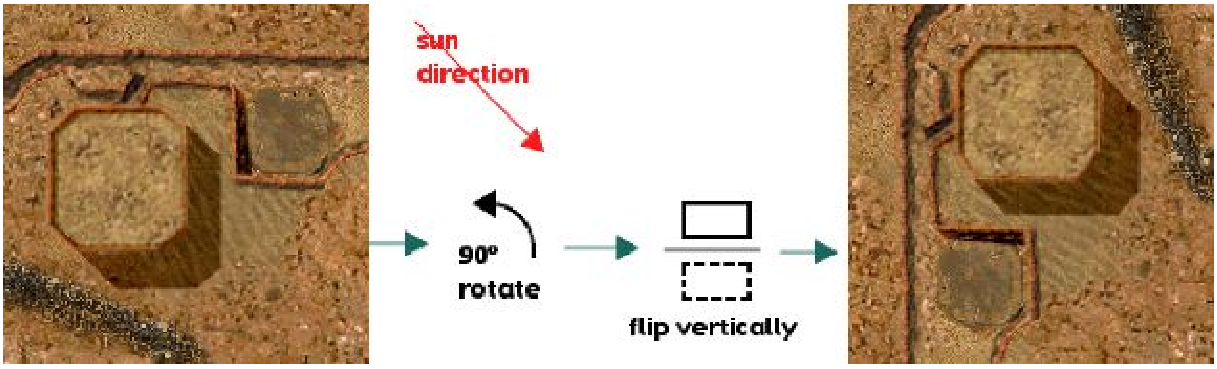
I have added a button in the "Single Actions" panel. Pressing the "Generate plain map data file..."-button will bring this window to front:

You must select the format of the new file you want to create: "CC2", "CC3" or newer. And you must determine the map's size (counted in deployment tiles). Maximum is 255x255 tiles, which might be too large for any CC game. Close Combat Marine (CCM, released on a CD inside the USMC Gazette magazine in August 2004) largest maps (BIG maps) are 160x40 deployment tiles large (as reported on the forums). Be warned: creating such large data files takes about 5 minutes on a 500 MHz G3. In addition you can determine the default elevation and the terrain element value (here: ranging from 0 to 8, which are valid values for all games from CC2 to RtB; and some special gimmicks for creating my AfrikaMod maps).

The v3.09 / v4.00 Update – rotating a map's data file

The problem: you have made a map, everything is ready. Then you encounter that the orientation of the map is wrong. In CC2 maps work best when they are oriented horizontally. If you have a map that is higher than its width, team deployment may become difficult.

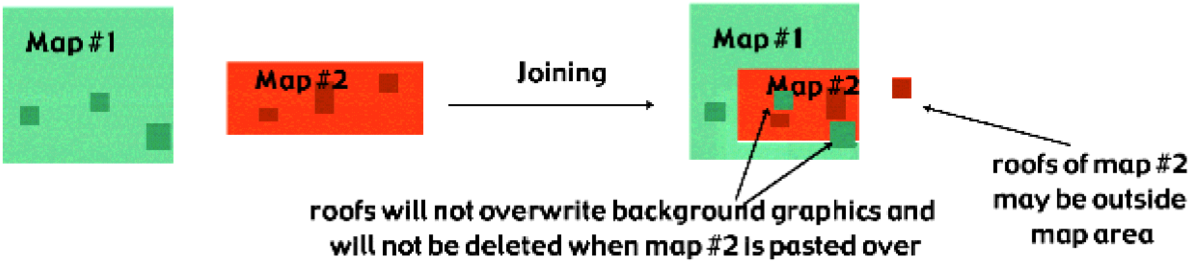
Now look at the map's sun direction. The sun-light on all CC maps is shining from the upper left corner of the map. All shadows are oriented bottom-right. If you rotate the graphics 90° counterclockwise and then flipping them vertically, the shadows are still correct, but the oritentation of the map is totally new:



Doing this in your favourite graphics editor is simple. I have added a new tab panel called "Rotate" where you can perform this action for the map's data file (Map### / *.txt), the map's background graphics file (BGMap### / *.bgm), the map's roof file (Roof### / *.rfm) and the map's bridge file (Bridg###)! The routines will detect the input file format by interpreting the file's header ID. So it requires that the input files are encoded correctly.

The v4.00 Update – Mooxe's request for map joining

The task for the RtB/CCM map makers: joining several CC5 custom maps together to get one new big 4800x4800 pixel map. In v4.00 of my tool I have implemented this feature in a seperate tab panel "Join". Again the routines will detect the input file format by interpreting the file's header ID. Joining will work for map's data, background graphics and roof files! Some restrictions: the first map file selected will serve as the "background basic" (and this might be the greater one of the two files to be joined together), and the second file selected will be pasted over the "values" of the first one. Exception: roof file joining will not delete any entries of the first or the second file (so it can happen that some roofs might hang over or might be outside the area of the first file. Correction of the roof file can be done in the "Shrinking section"). Map datas and background pixels of the second file pasted in will be clipped off at the edges of the first file.


Map #1
Map #2
Joining
Map #1
Map #2
roofs will not overwrite background graphics and will not be deleted when map #2 is pasted over
roofs of map #2 may be outside map area

The v4.00 Update – editing the "stri"-resource of ABTF executable (MacOS version only)

I have implemented a seperate window to edit the "CC strings" stored in the resource fork of the CC2 executable program "A Bridge Too Far". Sorry to say that at the moment the tool will not show control chars (like CR) correctly. I have disabled editing such strings by default, but you can enable it. Remember that writing the datas back to file will patch the executable program. Ensure that you have a backup copy of your program before doing so. and as always you are doing it totally at your own risk!

The v4.03 Update – pasting roof images over existing BGM files

For a special purpose: you want to have the roof images out of a roof file (any version) pasted over the BGM background graphics (exterior and interior graphics). This will be a way to get later on a TARGA graphic of the background with interior graphics pasted over. In the "Join" section you will now find two buttons to get this result. The resulting file will be always a BGM file. You can convert this file as usual into a TARGA file.

For some custom CC5 maps you must adjust the roof file interpretation a little bit. Below the two buttons you will find a popupmenu where you can determine the amount of pixels to be added/omitted at the right side of the roof images to get proper results. All CC2 maps and original CC3/CC4/CC5 maps do not need to be corrected.

Next problem: CC3-RtB roofs are in most cases no rectangles. After converting a CC3-RtB roof file into CC2 format you will get its images always as rectangles, which might intersect each other. Using the "Roof" button in the "Convert/Shrink" section will load the additional graphics needed for this conversion from the *.bgm file residing in the same folder where the *.rfm file is located. If the roof images have intersections, you will get strange results with exterior graphics pasted over interior areas. To avoid this, do the following: prior to converting to CC2 you must create a seperate new *.bgm file with interior roof graphics pasted over the *.bgm file using the new introduced button in the "Join" section. Give this file the filename extension ***.intbgm** and rename the original *.bgm into ***.extbgm**. Then the converting routine will detetct these files automatically (if they are in the same folder as the *.rfm file) and will take the correct graphics for interior areas out of the *.intbgm and for exterior areas out of the *.extbgm file even for intersections.

The v4.13 Update – expanding map data files to max

In September 2006 I was asked if there is way to make maps beyond the already known map size limit. Because CC2Tools.exe does not hold much data in RAM at runtime the generating of maps is only limited by it's popupmenus' number of entries. So I replaced the existing popupmenus in the section "Map-Expand" and "Map-Join" by textfields. Now it is possible to expand existing maps to sizes like 120x120 deployment tiles and even beyond. The user is solely responsible to enter valid (positive) values not exceeding the 32-bit integer barrier!

As an example we will see what this tool can do to a usual 20x20 deployment tiles coming from CC5:

- The map's data limit is 2^{32} elevation tiles (elevation tile = 40x40 pixels) depending on the compiler used to make this tool (and to make it work in CC5/CCM or newer: depending on the compiler used to generate the CC engine ... and on the amount of RAM you have when playing the game). RAM is no limit when using the tool.
- The elevation tiles are counted from 0 to maximum possible integer value, in case of 32-bit compiler arithmetic 2^{32}.
- Assuming that it will be again a square, the map's maximum size can be therefore theoretically 65535x65535 elevation tiles = 21845x21845 deployment tiles. This is the maximum size this tool can generate or handle when expanding/joining maps.
- But there is another limit: if you are going to expand the BGM and ROOF/Bridge files you will face pixel oriented coordinates. Width/Height entries and roof coordinates are encoded luckily since CC3 as 32-bit integers. That will mean that the map's data maximum will make request for a 2621400x2621400 pixel background graphic. Such size entries/roof coordinates can be handled by the existing CC5 engine. A 2621400x2621400 pixel BGM coded (like TGA) in 16-bit color will make a file size of around 13743475920 bytes = 12799 GB. Such map monsters can be created by this tool as a giant big black hole if you press the "Expand BGM" button without thinking about what you are doing.

The v5.00 Update – text file modding

In Dec. 2006 it became necessary to give the PC-CC2 modders a fast tool to bring all text-based files to CC2's standard file format. The CC2 standard text-based file format is "all lines terminated with CR". To be compatible with the MacOS version of CC2 it is necessary to convert modified text files (AlOOB, AxOOB, Scenario ...) made on a PC using PC tools (WordPad ...). These tools will create/save text files with "all lines terminated with CR + LF".

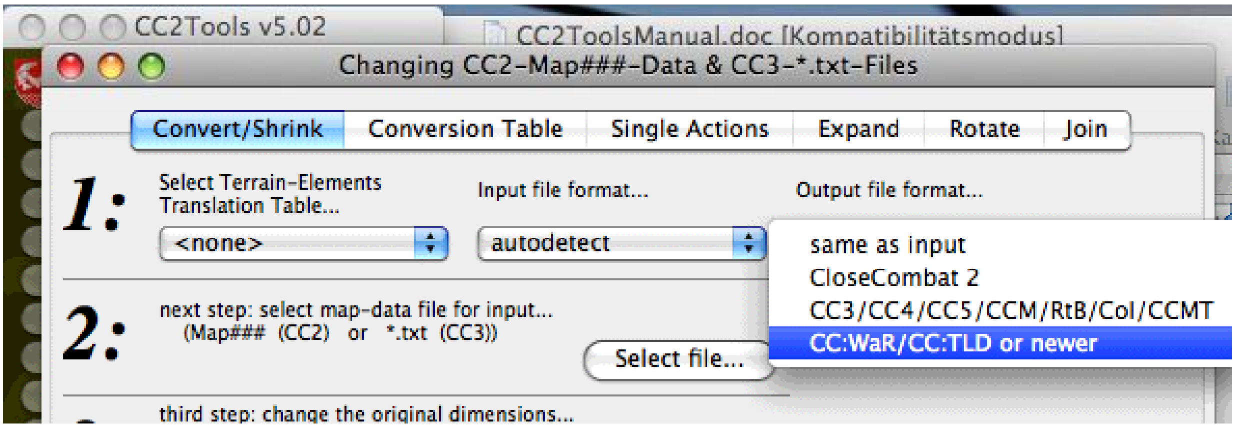
So I added the section "Adjust CC2 EOL-char". You can modify (in most cases: delete all line terminating LF chars) a single file or do this modifying to all files of your entire CC2 installation. You can select if you want to do this to all files without checking their names for being valid CC2 text-based files, but I recommend to use the selection which let you control the file types (recognized by their CC2 names). You can specify the filetypes you want to have scanned for LF chars. After selecting the root directory (in most cases: the folder "DATA") all files of this directory and all it's sub-directories will be scanned/modified (recursive function call). The modified files will be rewritten to their original location without changing their filenames and without making backups! So be warned: don't apply this functionality to graphic files or LOS-files! Make backups of your work before using such a massive file-manipulation!

The v5.02 Update – BTD-file shrinking/converting for CC:WaR/TLD

In July 2009 I added the shrinking/converting of BTD files in the following ways:

- shrinking CC:WaR/CC:TLD or newer BTD-files according to the settings made for the other map-files (this will not work with binaty CC4/5-BTD files!),
- converting CC2/CC3-Scenario files to CC:WaR/CC:TLD BTD-files (can be combined with shrinking),
- converting CC4/5-binary-BTD files to CC:WaR/CC:TLD BTD-files (can be combined with shrinking),

For this, I have added another output-file format option:



1: Select Terrain-Elements Translation Table... <none>
Input file format... autodetect
Output file format... CC:WaR/CC:TLD or newer

2: next step: select map-data file for input...
(Map### (CC2) or *.txt (CC3))

third step: change the original dimensions...

The tool cannot determine whether the input map's data file is CC4/5 or the new CC:WaR/ CC:TLD format (indeed, the file formats are identical). But the tool can automatically detect the file format of the input BTD-file. Sorry, an output option for writing old style binary BTD-files is not implemented. For more BTD-support see my new tool "BTD-Editor".

The other side

For MacOS-CC2-modders there are since some years much more tools available than in 2003: the excellent map maker "CCEdit" by Chris Ellens, Cpl_Filth's map-editor 3C.exe, my universal map- editor "5CC“ and several additional tools for the CC-series of games. Most of these tools are also available for PC-users. All tools helped me in making my CC2-Afrika-Mod.

This updated tool will help me and other CC5/CC2-gamers to have more CC5/CC2-maps on the harddisk. For PC users there a lot of more tools available: at the moment the largest tool collection can be found at JimmyD's toolshed:
www.wargamer.com/Hosted/CCJimmyD/index.htm
Another good source: CSO: www.closecombat.org, moved to www.closecombatonline.com
Since August 2004 a new source is online: www.closecombatseries.net

Manfred 'Mafi' Fischer Oct. 22nd, 2003
closecombat2@claranet.de
http://www.ftf.claranet.de/
http://closecombat2.fortunecity.com/
http://www.closecombat2.claranet.de/
http://www.dieppe.claranet.de/
http://www.afrika.claranet.de/
http://cc2revival.npage.de/
http://www.cc2.claranet.de/
http://www.mappa.claranet.de/