File: CC2Guide-UI_Intrface_OvData.PDF (former title: CC2Guide-UI_File_(MacOnly).PDF)
Format: PDF
Date: February 7th, 2001
Author: Mafi; closecombat2@claranet.de
Last revision: November 2nd, 2003 – v2: error revision, OvData# file description added

Close Combat 2 "A Bridge Too Far"

Patching the "UI"/"Intrface.cc2"-File + The OvData#-Files

What it is

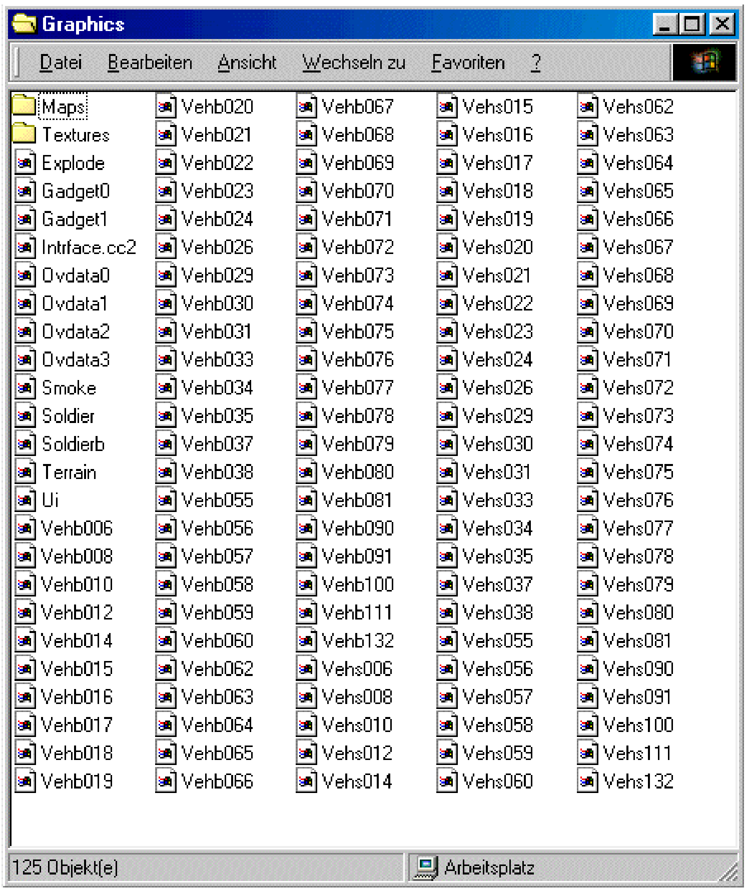
"Close Combat - A Bridge Too Far" (abreviated CC2, ABTF, CC2-ABTF) was the second game of the CloseCombat-series created by Atomic and presented by Microsoft to the Mac-community. It was also the last game of this series for the MacOS. The series was then continued by SSI (now by UbiSoft) for PCs only (up to day CC3, CC4, CC5). 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 / 10.3 (in Classic environment) as well. Later (localized) releases of CC2 were for PCs only. Some of the differences between the two installations will be described here (no at all complete):

Differences in storing graphical resources between the PC- and MacOS-version

Let’s have a look on the directory structure / file hierachy of the original CD:

PC-version:	**MacOS-version:**
Close Combat	**Close Combat**
Data	**Data**
Data	Data
Base	Base
Battles	Battles
Brief	Brief
Campaign	Campaign
Maps	Maps
Ops	Ops
Graphics	Graphics
Ovdata0	Ovdata0
Ovdata1	Ovdata1
Ovdata2	Ovdata2
Ovdata3	Ovdata3
Explode	Explode
Gadget0	Gadget0
Gadget1	Gadget1
Intrface.cc2	***Intrface.cc2***
Smoke	Smoke
Soldier	Soldier
SoldierB	SoldierB

Terrain	Terrain
VehB006	VehB006
VehB008	VehB008
VehB010	VehB010
VehB012	VehB012
VehB014	VehB014
VehB015	VehB015
...	...
VehB024	VehB024
VehB026	VehB026
VehB029	VehB029
...	...
Maps	Maps
Textures	Textures
UI	***UI***
Videos	Videos
Autorun	**Install (alias)**
Windows	**Mac**
DirectX	
Sounds	Sounds
Goodies	CC Sounds
CC2.hlp	
CC2.exe	
CC2hlp.dll	
CC2rsrc.dll	
Dsetup.dll	
Dsetup16.dll	
Dsetup32.dll	
Dxmedia.exe	
Eula.txt	
Setup.exe	Install
Setupenu.dll	
Tahoma	
TahomaBd	
Readme	**ReadMe**



Maps
Textures
Explode
Gadget0
Gadget1
Intrface.cc2
Ovdata0
Ovdata1
Ovdata2
Ovdata3
Smoke
Soldier
Soldierb
Terrain
Ui
Vehb006
Vehb008
Vehb010
Vehb012
Vehb014
Vehb015
Vehb016
Vehb017
Vehb018
Vehb019
Vehb020
Vehb021
Vehb022
Vehb023
Vehb024
Vehb026
Vehb029
Vehb030
Vehb031
Vehb033
Vehb034
Vehb035
Vehb037
Vehb038
Vehb055
Vehb056
Vehb057
Vehb058
Vehb059
Vehb060
Vehb062
Vehb063
Vehb064
Vehb065
Vehb066
Vehb067
Vehb068
Vehb069
Vehb070
Vehb071
Vehb072
Vehb073
Vehb074
Vehb075
Vehb076
Vehb077
Vehb078
Vehb079
Vehb080
Vehb081
Vehb090
Vehb091
Vehb100
Vehb111
Vehb132
Vehs006
Vehs008
Vehs010
Vehs012
Vehs014
Vehs015
Vehs016
Vehs017
Vehs018
Vehs019
Vehs020
Vehs021
Vehs022
Vehs023
Vehs024
Vehs026
Vehs029
Vehs030
Vehs031
Vehs033
Vehs034
Vehs035
Vehs037
Vehs038
Vehs055
Vehs056
Vehs057
Vehs058
Vehs059
Vehs060
Vehs062
Vehs063
Vehs064
Vehs065
Vehs066
Vehs067
Vehs068
Vehs069
Vehs070
Vehs071
Vehs072
Vehs073
Vehs074
Vehs075
Vehs076
Vehs077
Vehs078
Vehs079
Vehs080
Vehs081
Vehs090
Vehs091
Vehs100
Vehs111
Vehs132

There are no differences between PC- and Mac-version in storing or using the graphical resources in the files "Gadget0", "Gadget1", "Explode", "Ovdata0" – "Ovdata3", "Smoke", "Soldier", "SoldierB", "Terrain" and "Vehb006" – "Vehs132". Only the file "UI" has in the PC-version the size 0. It is a special file for the Mac-version. The filesize 0 occures because all of its resources are stored in the resource-fork and the data-fork is left blank. Viewing a hybrid-CD under a PC-operating system will only show the size of the data-fork of a MacOS-file. Viewing the same part of the CD on a Mac will show the real filesize: 10.2 MB. On the other hand the file "Intrface.cc2" is not used by the Mac-version of CC2, but can be found in the Mac.version of the CD. Also the sound-files are special to each version. Special to the PC-version are also all files in the folder "Windows". The file "CC2rsrc.dll" is therefore a special file for the PC-version.

Using of graphical resources in CC2

Most of the graphical resources in CC2 are used as sprites and are stored in files that are not different between the two installations. In the PC-version of CC2, greater pictures (for example the background-graphics (intro-picture, background-pictures for the menus, sector-maps ...)) are stored in the file **"Intrface.cc2"**. In the Mac-version of CC2 this graphical resources are stored in the resource-fork of the file **"UI"** (be carefull: this file has no data-fork) and can be viewed / manipulated using Apple´s "ResEdit 2.1.3" (can be found at: www.apple.com). The PC version of Close Combat 2 takes this resources only from the file "Intrface.cc2", the pictures here are stored uncompressed and encoded in Little Endian. Under MacOS the game takes them only from the file "UI" and they are stored as compressed PICT-resources (Big Endian encoding). Both files resides on the CD and are accessible under MacOS (see above). Due to the compression of the PICT-resources the file "UI" is smaller than "Intrface.cc2".

The file "UI" of the Mac-version

This file has only a resource-fork and can be viewed / manipulated using Apple's "ResEdit 2.1.3". It contains four groups of resources:

- grct
- PICT
- ppat
- TMPL

The "grct"-resources

- grtc-Resource-ID 129 "UI gadget rects", Size = 3954
- grtc-Resource-ID 1000 "UI gadget rects", Size = 1690
- grtc-Resource-ID 1001 "UI gadget rects", Size = 3954

Function not yet explored completely. The template for manipulating this resources in "ResEdit 2.1.3" is stored in the TMPL resource with ID=128. The grtc resources contain rectangle information (coordinates, sizes) for the position of buttons, pictures and other graphical elements of the user interface. Most of these graphical elements are stored in the files "Gadget0" and "Gadget1". Perhaps "grct" stands for "graphical rectangle".

The "PICT"-resources

In the Mac-version the background-pictures have all the size 800 x 600 pixels and the sector-overview-maps the size 314 x 491 pixels. All these pictures are stored in compressed PICT-resources (with the resulting filesize of 10.2 MB, which is less than the size of the PC-version's file "Intrface.cc2", where the pictures are stored uncompressed). They can be viewed or cut/copy/paste-manipulated using the "PICT"-template of "ResEdit 2.1.3". The numbers and the names of these PICT-resources in the file "UI" are as follow:

- PICT-Resource-ID 1000 "splash" --> "Intro-Picture" (800 x 600 pixels)
- PICT-Resource-ID 1001 "Command" --> "Command Menu Screen"
- PICT-Resource-ID 1002 "Debriefing" --> "Debriefing Menu Screen"
- PICT-Resource-ID 1003 "net" --> "Multiplayer Connection Menu Screen"
- PICT-Resource-ID 1004 "Details" --> "Debriefing Details Screen"
- PICT-Resource-ID 1005 "Video" --> "Background for presenting the videoclips"
- PICT-Resource-ID 1006 "Requisition" --> "Requisition Menu Screen"
- PICT-Resource-ID 1007 "Briefing" --> "Briefing Menu Screen"
- PICT-Resource-ID 1008 "Supply" --> "Supply Menu Screen"
- PICT-Resource-ID 1009 "Main" --> "Battlemaker Main Menu Screen"
- PICT-Resource-ID 1010 "Maps" --> "Battlemaker select by Map Screen"
- PICT-Resource-ID 1011 "Battles" --> "Battlemaker select by battle Screen"
- PICT-Resource-ID 1012 "Forces" --> "Battlemaker force edit Screen"
- PICT-Resource-ID 1013 "over" --> "Overview Map Netherland´s East" (314 x 491 pixels)
- PICT-Resource-ID 1014 "over arn" --> "Map Arnhem Sector" (314 x 491 pixels)
- PICT-Resource-ID 1015 "over nij" --> "Map Nijmegen Sector" (314 x 491 pixels)
- PICT-Resource-ID 1016 "over ein" --> "Map Eindhoven Sector" (314 x 491 pixels)
- PICT-Resource-ID 1017 "about" --> "Copyright Screen with logos of Atomic & Microsoft"

This is the same picture sequence as in the file "Intrface.cc2". The file "Intrface.cc2" has one more picture added: this additional (smaller) picture contain the menu-background pattern, which is stored in the file "UI" in the ppat-resource 128.

How to replace PICT-resources in the file "UI" on the Mac

You have to take a suitable graphic program (best example: the shareware "GraphicConverter" by Thorsten Lemke, demo can be found at "http://www.lemkesoft.com"), create a new file with filesize of 800 x 600 pixels at 16-bit color depth (because ABTF on the Mac is not able to handle millions of colours). Design now what ever you like in this new file. Save the new file as a resource, that means, place it into a resource-fork of a new file as a PICT-resource! (in GraphicConverter: File/Save as .../ with Format: "PICT in Resource" and Option: Extras: Add Resource "ON"). Now open the resource-fork of this new file using Apple's "ResEdit 2.1.3" program (http://www.apple.com), select the resource by clicking on the PICT-icon, select your new created picture by double-clicking the now presented preview and you will see the entire picture in 100%-scale. Use the "Copy"-command from the menu. Then open the file "UI" using again "ResEdit". Select the PICT-resources by clicking on the PICT-icon and select the picture you wish to replace by double-clicking it from the now appearing preview of all PICT-resources in "UI". Use the "Paste"-command of the menu and the new picture will be now in the file "UI" as a resource. Use the "Save"-command from the menu and you will be able to continue with other pictures the same way. If you will not use the "Save"-command, you will probably run into memory troubles.

In some cases you will be able to "Copy" and "Paste" PICTs from any graphic program into resources opened by "ResEdit". But it's not guarenteed. Be sure to use 16-bit graphics and to fit the correct size of the PICT-resource you are going to change.

The "ppat"-resources

Only one pattern-resource is stored in "UI":

- ppat-Resource-ID 128, size=2142

and cannot edited by a predefined template of "ResEdit 2.1.3". This resource contains the background color-scheme for the main battlefield screen-border and all scrollable menus. Size of the pattern is 128x128 pixels, in the original version with 1-bit colordepth! Background color is black, foreground color is olive green.

How to replace the old 1-bit colordepth ppat-resource by a new one

You can keep the size of 128x128 pixels, although it is not supported by a "ResEdit"-template. Take a suitable graphics program (like "GraphicConverter" by Lemkesoft) and create a new picture with the size 128x128 pixels and 16-bit color (or less colordepth). Draw anything you like as a background pattern. Save the picture as "ppat"-resource. GraphicConverter is able to do that (no saving options necessary).

Then launch Apple's "ResEdit 2.1.3" and open this saved file. "ResEdit" will open only the resource-fork and will present a lot of icon-, PICT- and much more resources. Select the ppat-resource, it will also have the ID 128. Copy it. Then open the file "UI", select the only ppat-resource and paste the new ppat in, overwriting the old one (if the IDs are not identical, you have to delete the old one manually, pasting the new one in and set the resource-ID to 128). Afterwards you have to save the file "UI". When you launch ABTF now, you will see the new created background-pattern in the scrollable menu-area, where you can select saved games or saved battlemaker files or for example in the boder and message/unit status area in the main battlefield screen. It works with 16-bit color-ppats as well as with 1-bit singlecolor-ppats or 4-bit- or 8-bit color-ppats!

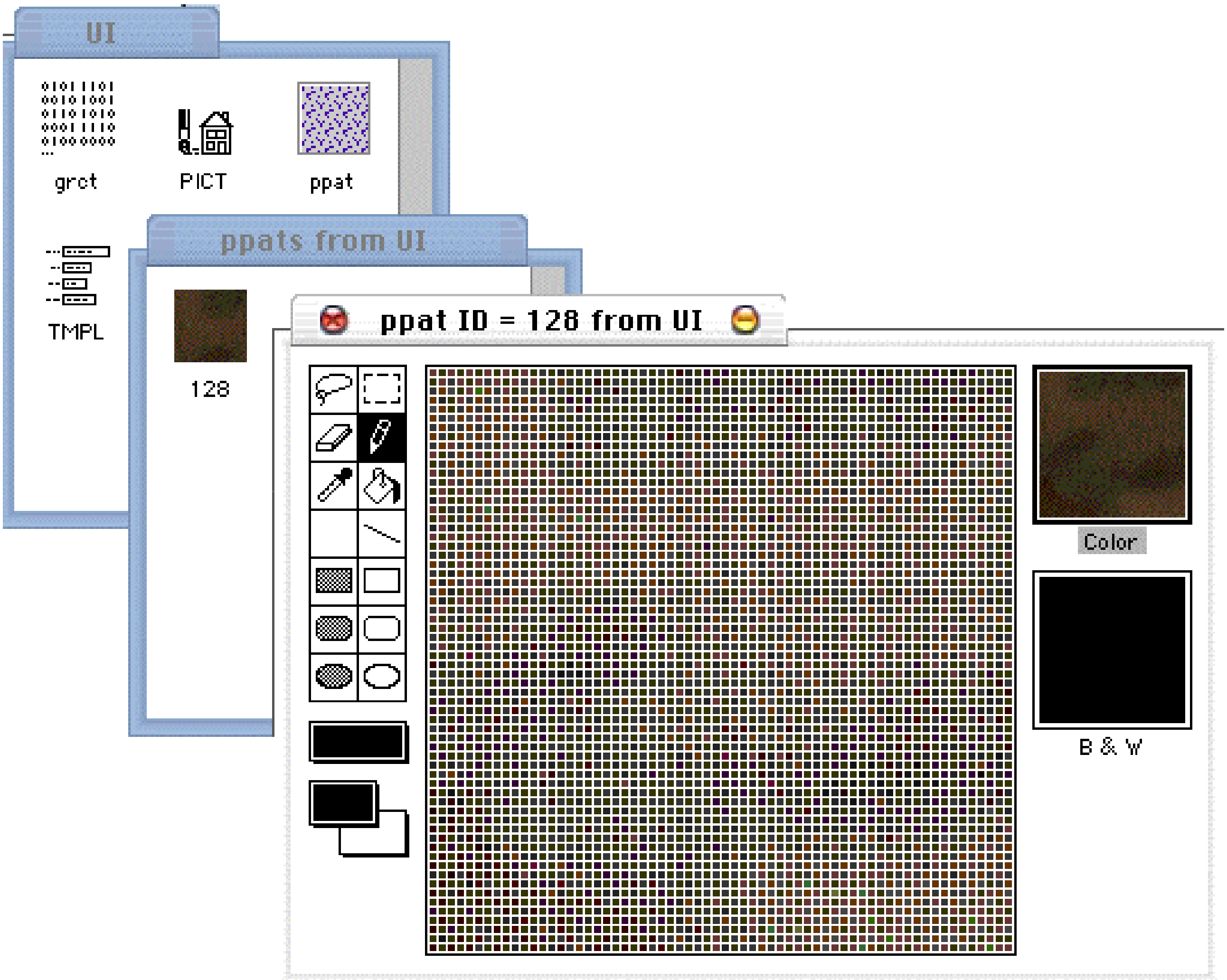
The original ppat

“Kreta”-like ppat

ppat for my Afrika-Mod

How to create a complete new ppat-resource without a graphic program

You can create a pattern resource using "ResEdit 2.1.3" without any further software. There exists a ppat-editor in "ResEdit". Open the file "UI" using "ResEdit". Select the "ppat"-resources, open them, select the only one ppat-Resource with ID=128 and delete it (Menu: Edit -> Clear). Create a new one. (Menu: Resource -> Create New Resource). The ppat-editor will open with the size 32x32 and colordepth 1-bit.



Picture: Editing the ppat-resource of “A Bridge Too Far” using Apple’s “ResEdit 2.1.3”.

Change the size to what you like out of 8x8, 16x16, 32x32 or 64x64 pixels (Menu: ppat -> Pattern size...). Change the color to what you like out of 4 grays, 16 grays, 16 colors, 256 colors, Apple's icon colors or "recent colors" (Menu: Color). Design your pattern or paste a pict in and set the resource-ID to 128! Once you have finished, save the file "UI". Although "ResEdit"'s ppat-editor is limited, it is possible to store greater ppat-resources than the limit 64x64 pixels, and ABTF uses this functionality of Apple's resource-system. And ABTF will use any ppat-size stored in the file "UI", if the resource-ID is set correctly to 128. I have tested ppat-Resources with the size 64x64 and 128x128 pixels.

The "TMPL"-resources

There exists only one template-resource in "UI":
- TMPL-Resource-ID 128 "grct", Size=62

Function not yet explored. "TMPL"-template for manipulating in "ResEdit 2.1.3" available. The resource contains 7 label/type-combinations:
1) Label "Num"; Type "OCNT"
2) Label "Gadget"; Type "LSTC"
3) Label "Top"; Type "DWRD"
4) Label "Left"; Type "DWRD"
5) Label "Bottom"; Type "DWRD"
6) Label "Right"; Type "DWRD"
7) Label "" (empty); Type "LSTE"
It looks like a record-description for a rectangle in an enumerated list. And it is the recommended template for editing the grct-resources of "UI" when using ResEdit 2.1.3.

The file "Intrface.cc2" of the PC-version

This file is absolutely PC-born. It has a complete different data structure than all other CC2 files. And it has now header identification. All numbers and all pixel values are coded in Little Endian / Intel-style byte format (last byte first). The structure of this file is:

```
//LITTLE ENDIAN for Intrface.cc2
//not used entries are padded with 00hex
//offset-, width-, height-tables
LongInt                //number of entries (should be 19)
Array(100) of LongInt  //100 offset values for the picture datas,
                       //counted from top of file
Array(100) of LongInt  //100 width values of the pictures, counted in pixels
Array(100) of LongInt  //100 height values of the pictures, counted in pixels

//picture datas, uncompressed, 16bit per pixel
data                   //number of bytes per picture = width * height * 2
```

The pixels are encoded in a slightly different format than in TARGA graphics: I used to translate them from "Intrface.cc2"-pixel-format into TARGA-pixel-format by:

> ValueRead = read two bytes from Intrface.cc2
> TARGA-pixel = BitAnd(ShiftRight(ValueRead,1), &h7FE0) + BitAnd(ValueRead, &h1F)

The 19 pictures are stored here uncompressed (remember: in "UI" the pictures are stored in compressed PICT-resources), so "Intrface.cc2" is larger than "UI" (original filesize is 13.4 MB). The picture sequence is the same as in "UI":

- picture #00 --> "Intro-Picture" (800 x 600 pixels)
- picture #01 --> "Command Menu Screen"
- picture #02 --> "Debriefing Menu Screen"
- picture #03 --> "Multiplayer Connection Menu Screen"
- picture #04 --> "Debriefing Details Screen"
- picture #05 --> "Background for presenting the videoclips"
- picture #06 --> "Requisition Menu Screen"
- picture #07 --> "Briefing Menu Screen"
- picture #08 --> "Supply Menu Screen"
- picture #09 --> "Battlemaker Main Menu Screen"
- picture #10 --> "Battlemaker select by Map Screen"
- picture #11 --> "Battlemaker select by battle Screen"
- picture #12 --> "Battlemaker force edit Screen"
- picture #13 --> "Overview Map Netherland´s East" (314 x 491 pixels)
- picture #14 --> "Map Arnhem Sector" (314 x 491 pixels)
- picture #15 --> "Map Nijmegen Sector" (314 x 491 pixels)
- picture #16 --> "Map Eindhoven Sector" (314 x 491 pixels)
- picture #17 --> "Copyright Screen with logos of Atomic & Microsoft" (800 x 600 pixels)
- picture #18 --> "Menu background pattern" (128 x 128 pixels)

The "OvData#" files (both versions)

Why describing the "OvData#" files in this document? Because they have a direct relation to the file "UI". The function of the file **"OvData0"** is to store rectangle information for the position of all sectors, operations (abbreviated: ops) and battles together with the names and ID-numbers of the sector, ops, battle (nearly in the same form as they are stored in the base-file "Batnames", see separate guide). These rectangles are used to show the user the selectable sector/ops/battle-zone. The selected zone will be highlighted (this is done by Close Combat 2 at runtime). And as background graphic serves the picture out of the PICT-resource 1013 of the file "UI" or the corresponding picture out of the file "Intrface.cc2". The datas of the file "OvData0" and the picture of PICT-resource 1013 are used by Close Combat 2 when displaying the "Command" screen (this is the screen you will see after the intro screen or intro movie).

"OvData1" contains the datas for the rectangles of the Arnhem sector (corresponding to PICT-resource 1014), "OvData2" those for the Nijmegen sector (PICT-resource 1015) and "OvData3" those for the Eindhoven sector (PICT-resource 1016). Now you will understand why the sequence of the pictures in the files "UI" and "Intrface.cc2" must be the same. ABTF in the Mac-version can access the PICT-resources by calling them by their name or by their index number (I think the programmers at Atomic used the way by calling them by number, later on you will understand why). In the PC version of ABTF the program must trust the picture sequence in "Intrface.cc2".

In addition the "OvData#" files contain information for each sector how many operation this sector includes, for each operation how many battles this ops includes and for every battle if the battle will take place at a landing zone (LZ), and/or at a bridge and/or has a road to be defended. This items of the battles are shown in the "Briefing" screen after selecting a operation/campaign.

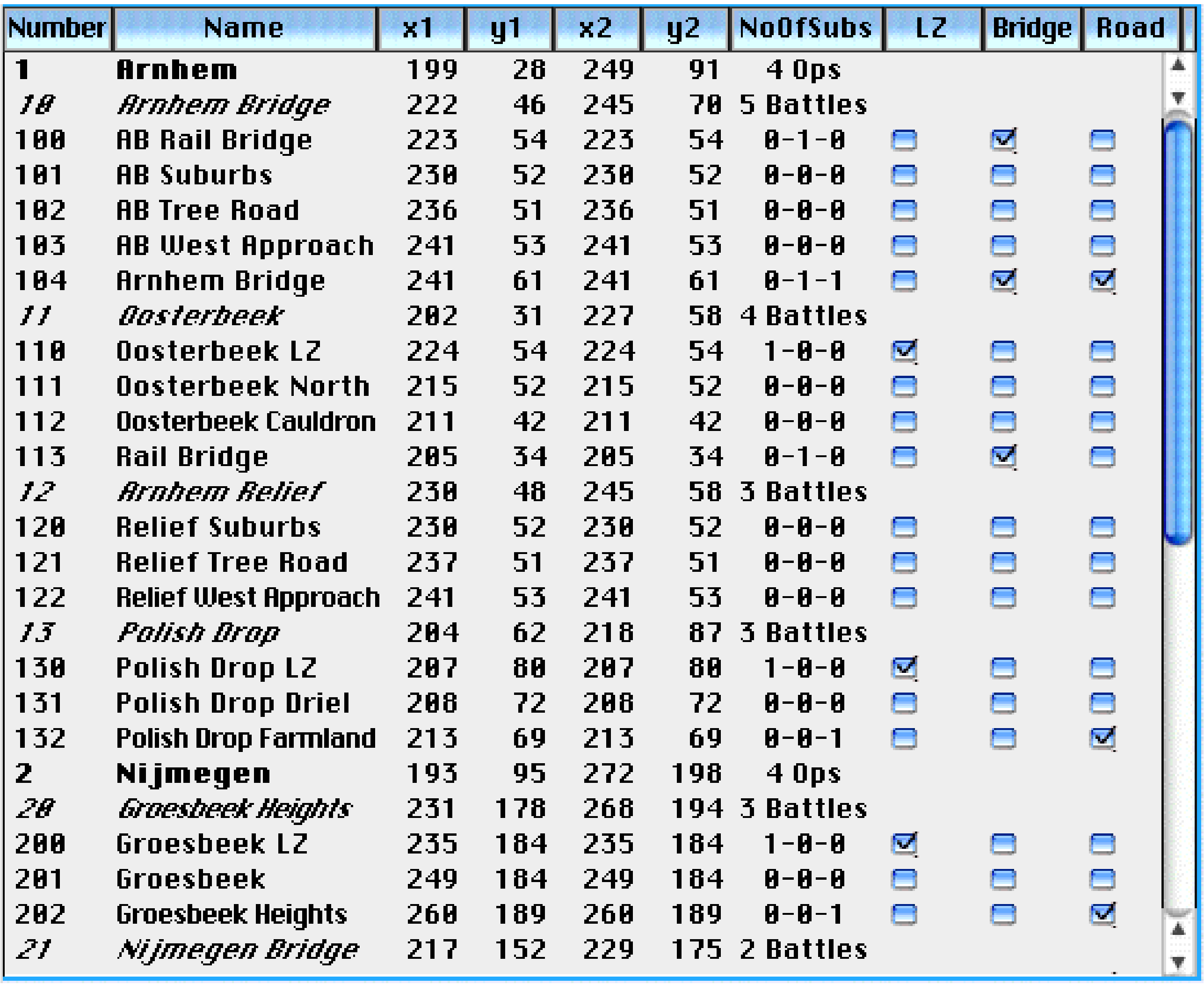
All "OvData#" files have the same hierachical data structure. "OvData0" contains datas for all the 3 sectors, the other three "OvData#" files hold datas for their corresponding sector. This is the general structure of the "OvData#" files: all numbers are stored MacOS-like in Big Endian style (high byte first):

```
//"OvData#"-file – BIG ENDIAN
//header, 16 bytes:
string(4) "OvDa"          //header ID "OvDa"
LongInt                   //unknown, value is always 1
LongInt                   //PICT-resource-ID to be used
LongInt                   //number of sectors

//for every sector:
LongInt                   //sector ID
string(50)                //sector name, zero-byte terminated, padded with A3hex
ShortInt                  //sector coordinate, upper left horiz. (x0), 2 bytes
ShortInt                  //sector coordinate, upper left vert. (y0),
ShortInt                  //sector coordinate, lower right horiz. (x1),
ShortInt                  //sector coordinate, lower right vert. (y1).
LongInt                   //number of operations in this sector

//for every ops in this sector
LongInt                   //ops ID
string(50)                //ops name, zero-byte terminated, padded with A3hex
ShortInt                  //ops coordinate, upper left horiz. (x0), 2 bytes
ShortInt                  //ops coordinate, upper left vert. (y0),
ShortInt                  //ops coordinate, lower right horiz. (x1),
ShortInt                  //ops coordinate, lower right vert. (y1).
LongInt                   //number of battles in this sector

//for every battle in this operation
LongInt                   //battle ID
string(50)                //battle name, zero-byte terminated, padded with A3hex
ShortInt                  //battle coordinate, upper left horiz. (x0), 2 bytes
ShortInt                  //battle coordinate, upper left vert. (y0),
ShortInt                  //battle coordinate, lower right horiz. (x1),
ShortInt                  //battle coordinate, lower right vert. (y1).
Byte                      //has landing zone (1) or not (0)
Byte                      //has bridge (1) or not (0)
Byte                      //has road (1) or not (0)
```

In the file "OvData0" all sectors have coordinates representing their position on the picture from resource 1013. In the other files "OvData1", "OvData2" and "OvData3" is only a single sector defined which has no valid rectangle coordinates (all values are set to zero). When modifying these files it is essential to set the sector/ops/battle Ids correctly. And it is essential to terminate the strings containing the names by a zero byte. As discussed in the guide concerning the base-file "Batnames" you can have up to 5 battles per operation. In the Mac version of th egame a sixth battle will be possible, but the PC version will crash when defining a sixth battle in an operation. The number of operations per sector migth be limited, but I have not tested it yet. Perhaps there maybe a limit of 5 ops per sector. And if you will look at "Ugur's method" of implementing additional maps the "Batnames" file can be expanded by battles from a 4th sector (see my guide concerning the "Batnames" file: CC2Guide-NewBattles&Maps.pdf).

As mentioned earlier, there is a relation between the "OvData#" files and the file "UI": in the header of each "OvData#" file is defined which PICT-resource from "UI" to use when displaying the rectangles in the "Command"- or "Briefing"-screen.

Modifying this entry in the header will result in displaying a different PICT-resource picture during runtime (tested in the MacOS version only yet). Example: I added a PICT resource 1018 to the file "UI" with a new graphic of the necessary size (that means: a real expansion of the file "UI"). And I changed the entry for the resource-ID in the file "OvData0" from 03F5hex to 03FAhex. Starting the MacOS version of CC2 will show the "Command" screen with my new graphic, ignoring the originally intended graphic from the PICT-resource 1013. That shows clearly: **CC2 calculates which picture to show in the "Command" / "Briefing" screen from the entry in the corresponding "OvData#" file.**



Number	Name	x1	y1	x2	y2	NoOfSubs	LZ	Bridge	Road
1	**Arnhem**	199	28	249	91	4 Ops			
10	*Arnhem Bridge*	222	46	245	70	5 Battles			
100	AB Rail Bridge	223	54	223	54	0-1-0	☐	☑	☐
101	AB Suburbs	230	52	230	52	0-0-0	☐	☐	☐
102	AB Tree Road	236	51	236	51	0-0-0	☐	☐	☐
103	AB West Approach	241	53	241	53	0-0-0	☐	☐	☐
104	Arnhem Bridge	241	61	241	61	0-1-1	☐	☑	☑
11	*Oosterbeek*	202	31	227	58	4 Battles			
110	Oosterbeek LZ	224	54	224	54	1-0-0	☑	☐	☐
111	Oosterbeek North	215	52	215	52	0-0-0	☐	☐	☐
112	Oosterbeek Cauldron	211	42	211	42	0-0-0	☐	☐	☐
113	Rail Bridge	205	34	205	34	0-1-0	☐	☑	☐
12	*Arnhem Relief*	230	48	245	58	3 Battles			
120	Relief Suburbs	230	52	230	52	0-0-0	☐	☐	☐
121	Relief Tree Road	237	51	237	51	0-0-0	☐	☐	☐
122	Relief West Approach	241	53	241	53	0-0-0	☐	☐	☐
13	*Polish Drop*	204	62	218	87	3 Battles			
130	Polish Drop LZ	207	80	207	80	1-0-0	☑	☐	☐
131	Polish Drop Driel	208	72	208	72	0-0-0	☐	☐	☐
132	Polish Drop Farmland	213	69	213	69	0-0-1	☐	☐	☑
2	**Nijmegen**	193	95	272	198	4 Ops			
20	*Groesbeek Heights*	231	178	268	194	3 Battles			
200	Groesbeek LZ	235	184	235	184	1-0-0	☑	☐	☐
201	Groesbeek	249	184	249	184	0-0-0	☐	☐	☐
202	Groesbeek Heights	260	189	260	189	0-0-1	☐	☐	☑
21	*Nijmegen Bridge*	217	152	229	175	2 Battles			

Picture above: analyzing the contents of the file "OvData0".

Summary

To change the look of the graphical user interface of CC2 in the both versions it might be sufficient to have the necessary pictures (for example as JPEGs). After saving them as PICT- or ppat-resources (on the Mac), they can be saved to the file “UI“ using Apple’s “ResEdit 2.1.3“, replacing the original resources of the game. If you want to do the same to the PC-version’s file "Intrface.cc2" you can use uncompressed 16-bit TARGA graphics and insert them into "Intrface.cc2" using ESCOBAR’s PC-tools "InterfaceX.exe" and "InterfaceI.exe" or my new released "CC2Tool", which is able to extract/rebuild the graphical content of this file under MacOS and Win. The coordinates of the sector/ops/battle-rectangles drawn over the overview maps in the "Command" and "Briefing" screens at runtime are stored in the files "OvData0" .. "OvData3". Editing these files can be done with a HexEditor or a new tool I will release soon.

MAFI

closecombat2@claranet.de
http://www.geocities.com/cc2revival/ - http://members.fortunecity.de/closecombat2/
http://www.closecombat2.claranet.de/ - http://www.cc2.claranet.de/
http://www.dieppe.claranet.de/