

EDITING THE TILES IN EAW

Version 2.0 by Stephen O'Leary (Pilot Officer Prune)

NEW IN THIS EDITION:

1. Improved a lot of the wording, took out typos etc.
2. Added fuller explanations of:
 - The terms used in this paper
 - The file formats, especially the file headers
 - How to convert between different file types
3. Added a section on editing the terrain elevation file.

CONTENTS OF THIS PACKAGE

None. The various files are now in a separate package.

SOFTWARE REQUIREMENTS

You will need the following:

1. Paulo Morais's CDFRW file-extraction utility.
2. *MapEdit*, by The Photon Effect, available free from the download section of:

<http://photoneffect.com/default.html>

3. A hex editor. I use XVI32, by Christian Maas, available free from:
<http://www.chmaas.handshake.de/delphi/freeware/xvi32/xvi32.htm>

The hex editor you use must support "find and replace", in cases where the replacement character is a null or empty string.

4. To create a map or use bitmaps: either a suitable blank map file, or a more advanced hex editor such as UltraEdit, which supports either macros or multiple string insertion. UltraEdit can be obtained from:

<http://www.ultraedit.com/>

- but it's shareware.

NOMENCLATURE

To (try to) avoid confusion, this paper uses the following terms:

- “World” or “EAW world”: the environment within which the player flies in EAW. It is not the same as:
- “The map”: the graphic screen seen by the player during briefings, while selecting mission parameters, or when consulting the in-flight map. The map is a picture (but not an exact representation) of the EAW world.
- “Layout”: refers to the placing of the tiles which make up the appearance of the terrain over which the player flies in the EAW world.
- “Elevation”: The height above sea-level of points in the terrain in the EAW world.

FILE FORMATS IN USE

1. Within EAW, the layout of the tiles is controlled by the file *EAW.tm*, which can be extracted from *data.cdf* using Paulo’s CDFRW utility. The layout is 640 tiles wide (west to east) and 320 tiles deep (north to south), for a total of 204,800 tiles.

EAW.tm consists of 204,800 bytes, each of which therefore represents one tile in the layout. Please refer to Dominique Legrand’s notes for full details of how it works.

2. The *MapEdit* program uses three file formats of its own:
 - (a) A *map file* with a name of the form ****.lcm*. This consists of a header section (approximately 44 bytes – the exact size seems to depend on the quantity of the data, and certainly depends on the name of the tiles file – see below) plus a certain number of data bytes. If the map is 640 by 320 as required for EAW, then there are 819,200 data bytes. This data is divided into 204,800 blocks, each consisting of four bytes. For EAW purposes, each block consists of:
 - (i) a single byte which codes the type and orientation of the tile visible at that point in the layout (refer to Dominique’s notes for the values of these bytes), followed by:
 - (ii) three zeroes.
 - (b) A *tiles file* with a name of the form ****.lct*. The exact makeup of this file doesn’t matter: all you need to know is that there is a file *EAWtiles.lct* which contains all the EAW terrain tiles in each of their four possible orientations.
 - (c) A *tiles bitmap* with a name of the form ****lct.bmp*. This contains a graphical

representation of the tiles, which MapEdit uses to depict them during editing.

3. Interestingly, a 256-colour bitmap of size 640 by 320 also uses one byte per pixel, with a header of 1078 bytes. It is therefore easy (and useful) to convert between the following three formats:

- 1 EAW.tm
- 2 MapEdit (*.lcm file)
- 3 256-colour .bmp file

IMPORTANT WARNING

When editing your layout, in whatever way, avoid using the first tile in the set (corresponding to the texture *Bnfield8.ter*). This is because its code in EAW.tm is 00. This makes it very difficult to carry out any of the file conversions described below, because there will be stray zeroes in awkward places!

BASIC OPERATIONS

To create a new map

To create an entirely new EAW world, you need an .lcm file which codes for 204,800 tiles. You could use the included file *EAW.lcm*, which has been made from *EAW.tm* as described in a later section; but a blank world file called *seablock.lcm* is also included in this package. It contains nothing but sea tiles in various random orientations.

Load one of these *.lcm files into MapEdit and edit away! The EAW tiles file will load automatically. Some notes on the actual use of MapEdit are given in a later section.

To edit an existing map

You need the .lcm file. Just load it into MapEdit and edit it.

If you don't have the .lcm file, see the next section.

FILE CONVERSIONS

Converting an .lcm file to a .tm file

In order to use your creation in EAW, you must make a .tm (EAW tile layout) file from the .lcm file used by MapEdit.

1. Load the .lcm file into your hex editor.
2. Remove the file header (hex: bytes 00 to 2B – but see below)
3. Check (using the Find function) that there are no strings of four zeroes (ie. 00 00 00 00) If there are, it means you ignored the Important Warning given above!
4. Replace all strings 00 00 00 with null strings (ie. remove them).
5. Save as a .tm file.

The file header: if you are using the EAW tile set supplied on my site, the header consists of bytes (hex) 00 to 2B. However, the header incorporates the name of the tiles file, so if you use a different set of tiles, the header may be a different size.

To check the size of the header, find the .lcm file in Windows Explorer, right-click it, and select Properties from the pop-up menu. Note the size of the file in bytes. The header size is that figure, minus 819,200 (the number of data bytes).

Converting a .tm file to an .lcm file

This is where you need an advanced hex editor. As editors differ in their details, I can only give a general guide here.

Basically you need to start with an .lcm file suitable for EAW use. The blank sea file *seablock.lcm* will do fine. Load it into your hex editor and copy the header – the first 44 bytes (hex: 2C bytes, ie. bytes 0 to 2B) - to the clipboard. Close the file, but not the editor.

Now load into the hex editor the map file you want to edit, which will probably be called *EAW.tm*. It should of course contain 204,800 bytes. You must use your hex editor to insert a string of three bytes (hex 00 00 00) after each existing byte. To do this using Ultra Edit:

1. Select File|SpecialFunctions|InsertString
2. The string to insert is “===” (three equals signs) and the increment is 1. Set the radio button for Start Count at Beginning.
3. Click OK.
4. When the process finishes (a few seconds), select Search|Replace
5. The string to be searched is 3D (hex code for the equals sign). The replacement string is 00.
6. Click on Replace All.

(The reason for this two-stage process is that for some reason I can't get Ultra Edit to insert strings of zeroes straight off.)

Now put the cursor at the very first byte of the file, and paste in the bytes you saved from *seablock.lcm*. This gives your file the correct header details. Save the file as [something].lcm, and load it into MapEdit to work with it.

WORKING WITH BITMAPS

Because a 256-colour bitmap has a similar structure to *EAW.tm* or a MapEdit file, a suitable bitmap can be used as the basis for an EAW tile layout. This will not prevent the need to edit tiles in MapEdit, but it will ensure that you get your basic geography sorted out right from the beginning.

It is also easier to convert between *EAW.tm* and a bitmap (or vice versa) than between an *.lcm file and other types.

One odd thing to note is that the bitmap is “upside down”. If you convert a .tm file to a bitmap and then open the bitmap, you will see that the geography is inverted. It is then easier to edit as a bitmap if you invert it into the correct position first. Don’t forget to re-invert it again before converting to a different file type!

Converting a .tm file to a bitmap

1. Using your hex editor, open the included file *EAW.bmp*
2. Select the first 1078 bytes (hex: 436 bytes, ie. bytes h000 to h435) and copy them to the clipboard.
3. Close the file, but not the editor.
4. Using the hex editor, open *EAW.tm*.
5. Paste in the saved bytes at the very beginning.
6. Save it as [whatever].bmp

The file should now open in any decent graphics program. Apply the palette *tileplan.pal* (PSP7 version) or *tileplan.act* (Adobe Photoshop version) to see (and edit) it in colours which reflect the types of tiles in use.

Converting a bitmap to a .tm file

Even easier. Just use your hex editor to remove the first 1078 bytes (in hex, bytes 0 to h435 inclusive) of the bitmap.

PRACTICAL APPLICATIONS

My *EAW.tm* for Japan and Korea was based on a bitmap. I started with a relief map downloaded from the internet. (This was a good map for me to use, because in Japan, land use closely follows the relief. Cultivation and urban settlement are found in the low-lying coastal areas, forests on the lower slopes of the mountains, and alpine terrain above

that. You might need to start with a different type of map for a different area of the real world.)

Having rotated the map slightly, and distorted the geography a little, to get the map I wanted, I then converted it to a 256-colour bitmap and applied the special EAW palette. This I then converted into an *EAW.tm* file, and from there to a MapEdit file as well.

I am now (September 2001) working on the MapEdit file, correcting details of coastlines and land use. Next, I will put in the cities, roads and rivers by hand.

USING MAPEDIT

MapEdit is a great little program, but it comes with no instructions. The following are some tips based on my experience of using it:

1. When you start the program, it always asks for details of the new map you want to create. Just keep clicking OK until the program window opens. Close the two windows (one map and one tile set), and load your .lcm file. The EAW tile set will load automatically.
2. The single black tile which you see near the top left-hand corner when you start MapEdit is caused by the act of opening the file (!). You will need to replace it with a suitable tile.
3. Alternatively, open MapEdit by double-clicking on the relevant .lcm file in Windows Explorer. For some reason, if you do this, MapEdit will start without inserting a rogue blank tile (!).
4. Use the View menu to zoom in and out, and turn the grid on and off.
5. There are two editing modes, controlled by the two buttons beside the top left-hand corner of the map display. The upper button turns on drag-and-drop mode, and the lower button turns on tile insertion mode.
6. In drag-and-drop mode, you can drag a box around a group of tiles, and then move it with the mouse. The usefulness of this is limited by the fact that it leaves a blank space behind (!). There does not seem to be any cut/copy and paste facility.
7. In tile insert mode, select a tile from the tile set. Each time you click a space on the map, that tile is inserted there. To delete a tile, right click it, or overwrite it with a different one.

EDITING THE TERRAIN ELEVATION FILE

Terrain elevation is controlled by the file *EAW16.hm*. Each tile is represented by two bytes in the order (low byte)-(high byte).

By inserting an extra byte before each pair, and adding a suitable header, an *.hm file can be rendered as a 24-bit bitmap.

In the case of the map for *Downfall*, I proceeded like this:

1. Make a 256-colour bitmap representation of the tiles.
2. In PSP7, use the Colour Replacer tool to reduce the number of colours used to six, representing: sea, coast, land (including roads and cities), forest, foothills (mixed forest-and-mountain tiles) and mountains. You do this as follows:
 - (a) Using the Colour Sampler (eyedropper) tool, select one of the shades used for that type of terrain. Select it as both foreground and background.
 - (b) Set the Colour Replacer tolerance to about 20.
 - (c) Use the Colour Replacer on the entire bitmap.
3. I subsequently added (by hand and eye) a seventh colour for snow-capped mountain-tops.
4. Convert the resulting file to a 24-bit (16 million colour) bitmap.
5. Using the Colour Replacer, replace the existing colours with new ones, according to the following table:

<u>Terrain type</u>	<u>RGB value (decimal) of new colour</u>
Sea	0,255,255
Coast	0,255,255
Land	0,1,1
Forest	0,20,20
Foothills	0,40,40
Mountain	0,80,80
Snowcap	0,120,120

Using these particular colours, with all the R (red) values set to zero, enables us later to remove one byte (the zero byte) from every set of three, leaving just those pairs of bytes that code for terrain elevations. Note that for this reason the sea and coast are initially set to (0,255,255) rather than (0,0,0).

6. Save the bitmap.
7. Using the hex editor, remove the bitmap header (bytes h000 to h35).
8. As a precaution, search for any pairs of zero bytes (00 00). If you find any, there's a problem somewhere! Go back to the bitmap and investigate.
9. Replace all zero bytes (value 00) with a null byte (ie. remove them). This reduces each group of three bytes to a group of two.
10. Now replace all bytes FF by 00. This sets the sea and coast to zero elevation.

11. Save the resulting file as your *.hm file.

This process sets the terrain elevations to the following values:

	<u>Decimal</u>	<u>Hex</u>	<u>Feet MSL</u> (approx)
Sea and coast	00 00	00 00	0
Forest	20 20	14 14	1925
Foothills	40 40	28 28	3850
Mountains	80 80	50 50	7700
Snowcaps	120 120	78 78	11550

NB: the MSL values are arrived at on the basis that one (hex) value of altitude is about 2.67 feet.

You can change these values by using different RGB colours (but remember to keep all the Red values to zero).

One potential problem is that the elevation value is actually set at the top left-hand corner of each tile. In theory this might lead to a mismatch on the north and west sides of hills; but in practice it seems not to do so.

Another problem with this process is that it produces large flat areas of land (albeit at different elevations). To get around this, manipulate the bitmap further, as follows:

1. At step 6 above, after saving the file, open up the additional files *20plus4.bmp*, *40plus.bmp* and *80plus.bmp*.
2. Using pattern-fill with the Flood Fill tool, fill the forest, foothill and mountain areas (respectively) with these files. This introduces some variation in height.

I don't do this with the fields (they need to be flat to put air bases on) or the snow caps (which I prefer to adjust by hand).

NB: it is a good idea to change the colour you are going to fill, into some other, bright colour like red or yellow before filling it with the pattern. This makes it easier to make sure you haven't missed a bit!

The pattern-fill files were made as follows (instructions for using PSP7):

1. Start with a 64 by 64 square of the base colour for forest, foothills or mountain.
2. Add three new raster layers.
3. Flood each layer with a modified version of the base colour. In *80plus.bmp*, each layer has a green value 8 more than the layer beneath it; in the other files, each layer has a green value 4 more than the layer beneath.
4. Set each layer (other than the base) to layer type "Dissolve".
5. Set the opacity of each layer as follows:

80plus.bmp and *40plus.bmp*: Top layer 4, second 12, third 36

20plus4.bmp:

Top layer 8, second 16, third 24

6. Merge the layers and save.

Table of available pattern-fill files:

	Green values	Opacities
20plus4	0, +4, +8, +12	100, 24, 16, 8
40plus	0, +4, +8, +12	100, 36, 12, 4
60plus	0, +4, +8, +12	100, 36, 12, 4
80plus	0, +8, +16, +24	100, 36, 12, 4
80plus4	0, +4, +8, +12	100, 36, 12, 4

North Africa

This differed as follows:

1. Initially the number of colours was reduced to 7, namely: sea, coast (including lagoons), lowland (including roads and cities, both in European and desert areas), foothills (mixed forest-and-mountain tiles), mountains, desert plateau (and European forest), and desert hills. Desert depressions (on the plateau) were filled with lowland colour.
2. The jebels (desert mountains) were then filled in so as to leave a fringe of desert hills colour (which also marked the Escarpment) on their lowland side (if any), and a fringe of foothills colour on their plateau side. This necessarily made the plateau lower than the foothills.
3. Two further colours were added for higher areas of mountains. A different colour was used in Greece, Crete and Turkey from the rest of the map, because the mountains there are considerably higher.
4. After saving as 24-bit, using the Colour Replacer, these nine colours were replaced with the following new ones:

<u>Terrain type</u>	<u>RGB value (decimal) of new colour</u>
Sea	0,255,255<<<
Coast	0,200,200<<<
Lowland	0,1,1<<<
Escarpment/desert hills	0,10,10<<<
Desert plateau/Forest	0,20,20
Foothills	0,40,40<<<
Mountain 1	0,60,60<<<
Mountain 2	0,80,80
Mountain 3 (Greece, Crete, Turkey only)	0,120,120<<<

5. As well as replacing all bytes FF by 00, replace all bytes C8 by 00 as well (the coastlines).
6. At a later stage, the elevations in each area were adjusted to reflect the local topography:

Sicily Etna rises to over 9000 ft and the northern mountain spine to about 5000. Most of the rest barely reach 3000 ft. Colours were therefore set as follows:

	<u>Decimal</u>	<u>Hex</u>	<u>MSL (ft)</u>
Forest	10 10	0A0A	960
Foothills	20 20	14 14	1925
Mountains	30 30	1E 1E	2800
Peaks	50 50	32 32	4800
Etna	100 100	64 64	9600

Mount Etna looked good, but the rest of the mountains were disappointing. Reverted to:

	<u>Decimal</u>	<u>Hex</u>	<u>MSL (ft)</u>	<u>Fill</u>
Forest	20 20	14 14	1925	20plus4
Foothills	40 40	28 28	3850	40plus
Mountains	60 60	3C 3C	5775	60plus
Peaks	80 80	50 50	7700	80plus4
Etna	100 100	64 64	9600	(None)

Aegean Some of the mountains rise to over 5000 ft. This includes Crete and Turkey. Used the following system:

	<u>Decimal</u>	<u>Hex</u>	<u>MSL (ft)</u>	<u>Fill</u>
Forest	20 20	14 14	1925	20plus4
Foothills	40 40	28 28	3850	40plus
Mountains	80 80	50 50	7700	80plus
Peaks	110 110	6E 6E	10500	110plus