



Mod Tools – Terrain Generator Overview

Table of Contents

<u>Introduction</u>	3
<u>Terrain Files</u>	4
<u>Start Building the Terrain</u>	5
<u>Create a Colormap</u>	6
<u>Make a Detail Map</u>	6
<u>Paint a Character Map</u>	8
<u>Paint a Foliage Map</u>	9
<u>Make a Terrain File</u>	10
<u>Build the Terrain</u>	11
<u>Make a .TPJ File</u>	11
<u>Build the Terrain</u>	11
<u>Pack the Terrain</u>	11
<u>View Your Terrain</u>	12

Introduction

Building a new terrain for the Delta Force - Black Hawk Down game engine can seem complicated at first, but once you understand the steps involved, you'll be able to add new terrain to your missions in no time.

The Terrain Generator tool released with the other mod tools is named `TrnGen.exe`. You can find it in the `ModTools` directory that you created when you installed the new mod tools and documents.

The terrain generator pulls together all the pieces you create in other programs such as Adobe® PhotoShop®. Before using the tool to build your new terrain, you need to prepare other files.

Note: You need to be familiar with Photoshop to create new terrains for the Mod.

Before you start, create a backup copy of your Black Hawk Down game directory, and the `ModTools` directory.

As with the other mod tools released for Black Hawk Down, the Terrain Generator tool is not supported. Visit NovaWorld (www.novaworld.com) and join the forum to discuss the mod tools with other gamers.

Terrain Files

The Terrain Generator tool requires several files to build a new terrain.

The mod tool release provides various sample files for generating terrain. The base name is `sample`, and the actual file names have an underscore extension to identify their purpose.

When you create your versions of these files, the file names must fit the 8.3 file naming format. Ideally, the base of your name will only be 6 characters long. That way, you can follow the naming convention of the other files (e.g. `sample_c.psd`).

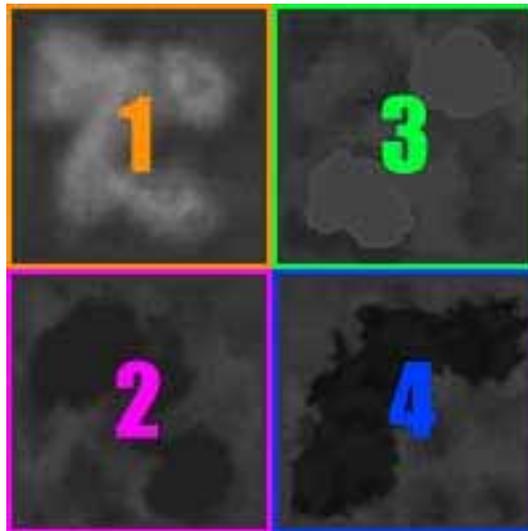
Here's a quick description of the various files. The upcoming sections talk more about them.

- **Colormap:** A 24-bit 1024x1024 targa file (e.g. `sample_c.tga`).
- **Depthmap:** The Terrain Generator tool creates a depthmap file from a 1024x1024 8-bit grayscale .RAW image. You can save your Photoshop image as a .RAW file, (e.g. `sample_d.raw`).
- **Detail texture map:** A 24-bit power of 2 square targa file. Named (e.g. `sample.tga`).
- **Character map:** An 8-bit 512x512 .PCX file with a specific palette (e.g. `sample_m.pcx`). Each palette index in the character map is used by the game engine to reference a sound or particle effect.
- **Foliage map:** An 8-bit 256x256 .PCX file with a specific palette (e.g. `sample_f.pcx`). Each palette index is used to reference a foliage type that is specified in the .TRN file.
- **.TRN file:** A text file that lists the various files above as well as the water height (if any) and the tile-rate for the detail map. You can start with the `sample.trn` file, and create a new one, for example `new.trn`, where you would rename 'new' to a 6-character long name.

Start Building the Terrain

This sequence of steps walks you through the process of building the required terrain files.

- Create a depthmap in a new 1024x1024 grayscale .PSD file, (e.g. `new_d.psd`).
- Each terrain is made up of four 512x512 quads that can be re-configured as long as their seams tile with one another. They cannot be rotated or flipped.



- Adjust the contrast and brightness. The brightness can be lowered anywhere between -65 and -90. The contrast can be lowered to a similar range. This will give you a nice undulating terrain base on which you can build specific features.

Tip: Sketch your mission on graph paper so you can extrapolate the depthmap based on each square in the grid.

- Next, take your mission design sketches drawings that mission design has done on graph paper and begin to create the depthmap based on their ideas. Brighter values represent the taller hills or mountains. Each pixel equals 1 meter on the map in the game. Each depth unit (0 -255) equals .5 meter of elevation change in game.

Tip: Place hills and riverbeds on different Photoshop layers. That way, you can use them as selections and create interesting colormap areas. This will make your depth and colormaps more cohesive. It will also help in the creation of the foliage map and character map files.

- Once you have created a depthmap you like, save it to `new_d.psd`.
- Flatten the layers and save it as `new_d.raw`. The **Header** value should equal 0.

Create a Colormap

- Create a colormap in Photoshop. Make a new 1024x1024 RGB file which you will save to `new_c.psd`.
- On a new layer, create a texture map that will be applied over the depthmap by using selections from the depthmap .PSD file you created earlier (e.g. `new_d.psd`). To create a texture map, you draw a region of varying light colors, where the lightest color is the highest. This textures map should follow the same quad tiling that you created in your depthmap.
- Once you have created at a colormap you like, save it (e.g. `new_c.psd`).
- Flatten the layers and save it in 24 bit mode (e.g. `new_c.tga`).
- Now you can make the shadow map for the terrain.
- Open `new_d.raw`. Duplicate the background layer twice.
- On the bottom layer, solid fill with RGB 128,128,128.
- Save your changes before moving on to the next steps.
- Now flatten the image, 'select all', and copy. We'll use this for the alpha channel in the next two steps.
- You can close your .RAW image now. Be sure NOT to save your changes. You were only using this file to create your shadow map.
- Open your colormap, (e.g. `new_c.tga`). Create an alpha channel and paste in the shadow map that you just copied. Re-save your colormap as a 32 bit .TGA file. This is your colormap.

Tip: Making new terrains requires many revisions. Along the way, sometimes very drastic changes need to be made to your depthmap. You may choose not to re-cook a shadow map for the colormap every time, but it's better to always make all changes to the depth and color maps in their respective .PSD files. This seems like a lot of steps but in the long run it works.

Make a Detail Map

Now you're ready to make a detail map to provide overlay texture for your terrain. The detail map will represent a highly detailed texture of a type of ground, such as dirt, stone, etc.

The detail map is a grayscale image of the ground. The detail map is a 24-bit .TGA file and it must be a square image with a power of 2. Size it to 512x512 and make it capable of being tiled. The `sample.tga` file will give you an example.

One way to create your own detail map is to take a digital picture of the ground outside your home, make your adjustments, and save it as the detail map. You can also use a sub-voxel map to give the illusion of greater detail on the terrain.

Save your grayscale detail map to your `ModTools` directory as the name of your terrain (e.g. `new_d.tga`).

Paint a Character Map

The character map tells the engine how to handle situations like sound in specific locations of your terrain. The character map file controls what particle effects are emitted from the ground when a bullet hits it or the sound a footstep makes as well. When in doubt, use index1, which is 'dirt'.

Before you start creating a character map file, you may want to wait until the visual aspects of your terrain are completed. Otherwise, you could initially position a mountain in one location of your map, then spend time in your character map identifying certain spots on your terrain as snow, but then later move your mountain to a different region. You'd then have to edit your character map again to move the snow and replace it with something else.

Until you're finished with your terrain layout, just use a copy of the sample file (e.g. `sample_m.pcx`). The solid brown color identifies everything on the character map as 'dirt', so you will at least hear dirt sounds when you run across the terrain.

Okay, let's edit the character map.

In your image editor, open your copy of the character map file (e.g. `new_m.pcx`). Size it up to 1024x1024 and under **Resample**, make sure **Nearest neighbor** is highlighted in the options box. (In Photoshop you can only copy selections from one image to another if they are the same size.)

Now copy a selection from your depth map into your resized character map image.

Tip: Use the **ColorTable** window of Photoshop to apply color indexes to different points of your character map. Remember that the indexes at the bottom of the window are in the reverse order (e.g. index 0 starts at the right).

When you run the terrain generator, it will read the character map you create and represent different terrain types based on the colortable palette indexes.

index	terrain type	index	terrain type
0	null	8	(not used)
1	dirt	9	mud
2	grass	10	ice
3	snow	11	(not used)
4	cement	12	rock or stone
5	sand	13	wood
6	packed dirt	14	metal
7	(not used)	15	(not used)

When you're done applying colors to represent specific index values, you need to return the character map back to the proper game size. In Photoshop (or your preferred image tool), resize the character map back down to 512x512 using **Nearest neighbor**. Save it into your `ModTools` directory.

Paint a Foliage Map

In the foliage map file, you can specify where foliage is placed automatically by the game engine.

However, just like we mentioned in the character map section, because your terrain will normally change quite a lot as you finalize it, you may want to wait until your terrain is done before creating the foliage map. Until then, use an existing file that you have renamed (e.g. `sample_f.pcx`). In that example foliage map file, the entire map is black because that colorable index (# 255) relates to "no auto-foliage". Once you've completed your land, you can create an appropriate foliage map.

The `sample.trn` file includes entries for various types of grasses and bushes. In this example, the `.3DI` file is an object file created by the NovaLogic Object Editor (OED). Refer to the **Object Creation Tutorial** for more information.

```
foliage
  graphic      mveglat.3di
  color_lower  0
  color_upper  2
  match        254
end
```

These are part of what we call the "auto-foliage" system. It is a very efficient way to give the terrain a type of ground cover; much more efficient than placing a large number of individual bushes. Similar to the character map image, each color palette index that you place on your foliage map corresponds to one of three types of auto-foliage defined in the `.TRN` file.

index 255 = null, draw no auto-foliage.

index 254 = draw whatever auto-foliage has the "match 254" line in the `.TRN`.

index 253 = draw whatever auto-foliage has the "match 253" line in the `.TRN`.

index 252 = draw whatever auto-foliage has the "match 252" line in the `.TRN`.

Note: For performance reasons, Black Hawk Down terrains do not use more than 3 types of auto-foliage on a single terrain.

Make a Terrain File

The .TRN file is the master file that tells the game engine what files to use for your terrain. Edit the `sample.trn` that was provided with the mod tools.

You will need to change several settings so that your own terrain files are found.

```
terrain_name          "Sample"
terrain_creator       "Novalogic"
polytrn_colormap     Sample_c.tga
polytrn_detailmap    Sample.tga
polytrn_polydata     Sample.cpt
polytrn_tilestrip    trntile10.tga
polytrn_ormap        Sample_m.pcx
polytrn_foliagemap   Sample_f.pcx
```

You need to make two more important changes to the .TRN file. These changes tell the terrain generator the size of your map, and specify how you want it to tile the four quads that you created in the **Start Building the Terrain** section.

If you only want the game to be 3 quad tiles wide, you set the `polytrn_sectorcount` to 3. In the `sample.trn` file, the terrain is set to 8 tile sectors.

```
polytrn_sectorcount  3          ; width of sectors
```

In this example, because you want a map that is 3 sectors wide, you need to tell the terrain generator how to construct the terrain, using the four quadrants. As long as you created your tiles with the same depth on the tile edges, you can organize the tiles in any format you choose. If your quadrants are drawn with water or mountains on an edge of the tile, you need to be more careful when you connect the tiles together.

```
polytrn_sectors  1    3    1
polytrn_sectors  2    4    2
polytrn_sectors  1    3    1
```

As mentioned in the **Paint a Foliage Map** section, you can add auto-foliage in the .TRN file.

Build the Terrain

Now that you have created all the necessary files, you use the terrain generator to create a depthmap file (.CPT) from your .RAW file.

Make sure that `TrnGen.exe` is in the same directory as the .RAW and the other terrain files that you created earlier in this process.

Make a .TPJ File

The .TPJ file is a text file that tells the Terrain Generator tool (`TrnGen.exe`) the configuration order for your terrain quads.

Create your .TPJ file by copying the `sample.tpj` file in your `ModTools` directory, and rename it for your terrain, (e.g. `new.tpj`). In your .TPJ file, change the path and file names to match your own.

This document doesn't describe how to change the lock sector values. If you do change them, make sure the values in the .TRN file match the values in the .TPJ file.

Build the Terrain

Now that you've prepared the other terrain files, you're ready to tell the Terrain Generator tool to build the final terrain file. The end result after using the terrain generator is a .CPT file, (e.g. `new.cpt`).

- Run the Terrain Generator from your `ModTools` directory, (`TrnGen.exe`).
- From the **File** menu, click **Open** and select your own .TPJ file.
- On the **Build** menu, click **Begin Build** to start the process of creating the .CPT file. This will take up to 20 minutes depending on the complexity of your terrain.

When you're done, you can delete these temporary files: *.TMS, *.TML, *.DEP, *.OBJ.

Pack the Terrain

Now you need to pack the various terrain files in the mod file.

To do this, you'll be adding these 5 files to the .PFF containing all your mod files (use your own terrain name): `new.cpt`, `new.tga`, `new_c.tga`, `new_f.pcx`, `new_m.pcx`.

You need to run the Pack tool for each file listed above:

```
pack MyMod.pff +new.cpt
```

If you create a batch file that contains a pack command for each of these files, it will make your life easier. The `packPFF` example batch file can be modified to suite your different packing needs. Also, refer to the **Pack tool Overview** document for more details.

View Your Terrain

Now that you've created the required files, run the Terrain Generator tool to create the final terrain file, and packed them into the .PFF mod file, you're ready to add the terrain to a Black Hawk Down mission.

You're ready to open the Mission Editor tool (MED) and add your terrain.

Note: The MED manual can be found in the `MANUALS` directory of your Black Hawk Down game CD. It's called `DFBHDMED`.

Follow these brief steps that will let you see your terrain right away:

- In MED, open the **Edit** menu and click **General Information**.
- In the **Mission Terrain** field, select your terrain name from the menu, and click OK to close the window. Your terrain should appear.
- Right-click on the terrain, and check **Insert Mode** in the menu.
- Left-click and choose **Markers** in the left column.
- Choose the **start, player** object and click **OK** to close the window.
- Click **Save and Export** from the **File** menu.
- Enter your mission name and exit MED.
- Use the Pack tool to add your new mission .BMS file into your mod .PFF file, (e.g. `pack MyMod.pff +mission.bms`).
- Copy your mod file to the `mod` directory in the Black Hawk Down game directory.
- Launch Black Hawk Down, passing it your mod file, (e.g. `dfbhd /mod MyMod`).
- Go to Single Player and select your mission name from the list.

Congratulations! You should be walking around on your own terrain.

From here you can improve your terrain through trial and error. Some designs don't work right away and need to be evaluated for both game play and aesthetic value, but once you've gone through this process, it becomes easier.