

POG Modding: Modify objects on the map

Inhaltsverzeichnis

- [1 HOW TO MANUALLY EDIT POG FILES](#)
- [2 POG FORMAT STRUCTURE EXPLAINED](#)

NOTE: This (outdated) guide explains the old way of manually editing POG files, if you don't want to mess around with HEX values you can use the POG tool for faster editing with decimal values instead: <https://skybreaker.eu/download/how/HoW-PogEditor.exe>

1 HOW TO MANUALLY EDIT POG FILES

1. You need the PC version, do NOT install your game it in C:/Programm Files, I recommend C:/Hogs of War
2. You need to download 2 programs:
 - A Hex Editor, I recommend Hex-Editor MX or Hex Editor Neo.
 - Notepad++
3. Backup the whole map folder in your hogs of war directory (and also any other files you want to modify)
4. In the map folder you will find all the maps you can edit, the first mission is called "ESTU.POG" ([here is a list of all the missions](#))
5. Open ESTU.POG in Notepad++
6. If you want to remove one object on a map, you need to erase the WHOLE block from the object name until the next object name ! (see screenshot)
7. You can add an object in the exact same way, you need to copy & paste the WHOLE object block. It doesn't really matter where you paste it, but it shouldn't be at the very beginning, nor at the very end, just paste it right in front of another object block. In Notepad++ there are special functions to copy the binary content, if you use regular copy/paste it won't work, you need to use: Edit --> Paste Special --> Copy/Paste Binary Content
8. You must not add spaces/return or any additional characters to the file, otherwise it will crash. You literally can only copy & paste content from various map files.
9. Once you have pasted the desired objects into your POG file you can edit the objects yourself. To do this, you have to open the file in the Hex-Editor MX.
10. The file is now shown in hexadecimal format, each pair represents one value that has a meaning in the game. Now the struggle starts, you basically need to find out what the meaning of all those pairs is and what other values the pairs can accept. You can do that by manipulating the values and check if it changes something ingame or you could also check the other mission files and see if other values are used there (or if similar values are used in the same spots).
11. You don't have to do all of the work, you will find a nearly complete list below. However, in case you figure out new values, please share them with me !

(image not found)

If you want to remove the GR_ME object, you need to remove everything till the next object (GR_HV).

You can also copy the whole block starting from GR_ME and ending right before HV_ME, then you paste it again directly in front of the HV_ME. This way you will have two times the same GR_ME object.

2 POG FORMAT STRUCTURE EXPLAINED

The **first two bytes** of the POG files contains **the number of objects** in the mission. Then in all the following bytes these objects are described.

In this example I show the object HV_ME (it's a pig), but my findings will work on any object. The data size of all the objects is exactly the same (94 bytes) you can edit each of them and manipulate how the objects behave in the game in some way.

[111-hex-format-png](#)

Image not found or type unknown

Black - Name

Spoiler anzeigen

Dark Green - X|Z|Y Precision

Spoiler anzeigen

Green - X|Z|Y Coordinates

Spoiler anzeigen

Beige - Object ID

Spoiler anzeigen

Silver - Rotation Precision

Spoiler anzeigen

Grey - Rotation

Spoiler anzeigen

Dark Yellow - Unused ?

Spoiler anzeigen

Red - Pig Rank ID / Parameter 1

Spoiler anzeigen

Terracotta - Collision

Spoiler anzeigen

Light Purple - Advanced Collision Settings

Spoiler anzeigen

Yellow - Object Value

Spoiler anzeigen

Brown - Multiplier for bigger values

Spoiler anzeigen

Purple - Spawn Settings

Spoiler anzeigen

Violet - Group Value

Spoiler anzeigen

Turquoise - Script Event

Spoiler anzeigen

Dark Blue - Script Group

Spoiler anzeigen

Blue - Script Parameter 0

Spoiler anzeigen

Light Blue - Script Parameter 1

Spoiler anzeigen

Mint Green - X|Z|Y Droppoint Precision

Spoiler anzeigen

Yellowish Green - X|Z|Y Script Location

Spoiler anzeigen

Orange - Start Settings

Spoiler anzeigen

Blue Grey - Player Type

Spoiler anzeigen

Gold - Additional Settings

Spoiler anzeigen

There are 17 additional bytes on the 4th line (from 0x41 to 0x51). Those values are used for advanced scripting, a tutorial will follow.

If you find any other values or discover some new then please share it with us (here or on the [Hogs of War Discord](#))

HUGE THANKS to Valera & Mikhail for discovering most of those informations ! 