Modding Guide for Mount&Blade Warband

This guide contains among other things

- Setting up the Module System
- Module System Syntax and Usage
- Documentation Module System
- Documentation module.ini
- Game Engine Settings, Formulas and Info Tidbits
- BRF Resource Management

last edited at: 29. Dezember 2021

Abstract

Here a little text about the content of the modding guide (Collections of informations and tutorials) and what won't be content here, at least not en detail (3d-modelling, world map and scening).

Credits for those who helped at writing down this guide and/or provided informations at the forum which have been integrated (alphabetic order):

Abhuva, Amman d Stazia, Antonis, Arch3r, Baskakov_Dima, bryce777, burspa, Caba'drin, cmpxchg8b, Colt, Dargor, Dalion, Darth Mongol the Unwise/MongolFromForest, Dj_FRedy, Doggy, dstn/Dustin/SupaNinjaMan, dunde, Dusk Voyager, Earendil, fisheye, GetAssista, HokieBT, InVain, ithilienranger, jacobhinds, jik, K700, kalarhan, Khamukkamu, kraggrim, kt0, La Grandmaster, Lav, Leonion, Llew2, lllew, Lumos, MadocComadrin, MadVader, Mammoet, Masterancza/Garedyr, maw, Mirathei, mtarini, Neophyte, Norwood Reaper, NPC99, Pitch, rabican, Ritter Dummbatz, _Sebastian_, Seek & Destroy/Seek n Destroy, Sicnarf, silverkatana, Sinisterius, smiling_cataphract, Somebody, SonKidd, Specialist, Swyter, The_dragon, The Bowman, Thorgrim, Veledentella, Vetrogor, Vjeffae, Waldzios, Watakushi, Winter, WookieWarlord, xPearse.

Table of Contents

1 Introduction[1]

What actually is “modding“?

As you probably already know, the word “modding“ comes from “modifying“, and “mod“ can mean either “modification“ or “module“. But what the word means doesn't really matter. Shortly put, “modding“ means modifying something. In this case you're going to modify the game Mount&Blade or one of its derivatives. Modding actually is programming, but on a basic level, so if you have experience in programming, you shouldn't have any problems in understanding how to mod.

And how does modding in M&B work?

To understand this, you need to understand a few basic, easy concepts of programming. So, open up your Mount&Blade\Modules folder. There is the Native mod that ships with the game, and any mods you may have downloaded and played. Note that I say “Mount&Blade“, but this can be any of the games you want. We'll presume we're talking about Warband, so your folder will be “Mount&Blade Warband“. Or whatever.

Take a peek inside the Native folder. You'll see a bunch of text files and some folders. Open a text file, let's say “presentations.txt“, and scroll down a bit... Unintelligible, isn't it? Except for some text here and there, the rest of it is digits. This is what the game reads, and this is how a mod looks like when it's done.

To make a mod, you can directly edit the text files, although this is a boring, tedious, error-prone job. My advice: don't do it. Of course, I also did it when I first started out (not to make a mod, but to give myself some über stats and... um... yeah). The other way is to use the “Module System“, which is a set of script files (with understandable language and much less numbers!) that are compiled using Python, a programming language. It is important for you to know that although the files from the Module System (I call it ModSys for short) have the .py extension (Python file), the ModSys itself is not written in Python, and (for 99.99 % of your needs) you don't need to know it (yes, you can use it for complicated stuff if you know it.). The ModSys uses something way easier - just a bunch of “operations“ which do the job. And after you mod these .py files, the real pythonic compiler compiles (translates) these files and creates the .txt files you saw earlier, which the game can read.

The M&B “language“ does not have a fixed name. I call it Mount and Blade Language - MABL (pronounced “Mable“), but many people seems to prefer M&B Script. This does not matter at all, though. You can call it “Ni!“ and it would still work. (If you have a shrubbery... lol)

[1] Taken from *The Ultimate Introduction to Modding* by Lumos.

All right, I got it. What do I need to do to modify my game?

It depends on what you want to do. If you want to tweak stuff, add some new items, or some troops, or change existing stuff, and you want to do this for personal use (mostly), you can edit the text files - either by hand (NOT recommended) or by using any suitable modder-made application from the Unofficial Editing Tools board. If you have a serious project in mind, you'll need the ModSys. Also, keep in mind that compiling the ModSys overwrites any changes done in the .txt files.

There are, in my humble opinion, three main categories of modding. These are Coding (writing scripts for awesome stuff, creating new game features) - which means working with the ModSys, Modelling (making the visible stuff: items, houses, anything that is a 3D model) - which is mostly not working with the ModSys. However, to add your stuff to the game you'll still need to know how the ModSys works; and Texturing (giving the 3D models an actual look by using drawings) - of course, most modellers are texturers, but you can include any other type of 2D art (and the game uses that too) in this category. ModSys knowledge not required, but recommended.

I'm beginning to think that you're going to talk about this ModSys even more.

You're right. I'm a coder. My 3D modelling skills are... hm, pre-intermediate at best, and my texturing skills are minimal. (I'm sorry, but I can't draw with a mouse. I draw reasonably good on paper, but I suck hard with the mouse.) And, as a coder, I'm going to focus more on the coding side. Of course, I'll talk only about the basics, which everyone should know.

And what if I don't want to be a coder?

Very good, but you could still learn something that you'll use someday in the future. If you're looking for software, you can use anything that exports as an .obj/.smd for models and anything that can edit .dds files for textures (or use the DDS Converter for that).

Software for 3D modelling includes Wings3D and Blender (freeware), 3DSMax, Maya, ZBrush (paid) and so on and so forth. Software for texturing includes GIMP, Paint.NET (freeware), Photoshop (paid) and others. Look around the tutorials in this forum to understand more game-specific stuff about modelling/texturing. This will be a very good place for you to start.

Kudos to you for reading these sentences. Of course, it's presumed you've read through the whole preamble. Now grab yourself a beer or something... if you haven't already.

- *Lumos*

2 Localization and Modularity[2]

Usually, mods have their own folders with their own files as you will learn soon but files for Native are nowhere to be found. Try looking a level or two up, directly in the game folder. CommonRes (a Resource folder), languages, music, Sounds, and Textures are identical to their mod folder counterparts, but are common to every mod. This however doesn't mean that they will be fully used in every mod (pretty much only in Native). To make a mod fully independent, you've got to modulize it. This includes localizing the files by copying files from the common folders to the mod specific folders. Next, you must get the module to read from the right directory. More information on this below.

Mod Folders and Files

What exactly constitutes a mod? A mod essentially is a bunch of files in a bunch of folders. If you change the files, you create a new mod. Therefore, it makes sense to learn about what file types go in each folder and which software to use for what. This is covered below.

All of these files and folders can be found here, where Mod is the name of the mod.

```
C:\Program Files\Mount&Blade (Warband/With Fire and Sword)\Modules\Mod
```

Languages

Use Notepad for the CSV files. Simply localize to modulize, otherwise it reads from the common languages folder. The CSV files contain various text found in-game such as hints. You may only edit the text after the | in each line. To add a new hint, add a new line starting with hint_#| in hints.csv. See module.ini for more information.

Music

Use Audacity for OGG, WAV, and MP3 files. To modulize, use the music track flag mtf_module_track for all applicable tracks in module_music.py within the module system. The folder contains audio specifically for background music.

CommonRes/Resource

For BRF files. Use OpenBRF in conjunction with Wings3D, Blender or another programme of your choice. To modulize, use load_mod_resource or load_module_resource instead of load_resource for

[2] Taken from *Modding Tools and Files* by Pyrate.

each applicable BRF in module.ini. This folder contains graphics such as skeletons, animations and meshes. Skeletons provide the animations for rigged meshes. Meshes give objects in the game their three-dimensional form. These are what materials are UV-mapped to or wrapped around. The folder also contains data for materials, textures and shaders. To rework skeletons, animations or meshes simply export then import them into Wings3D, Blender or another programme.

SceneObj

Use the in-game editor for SCO files. They are always local and modulized. The folder contains scene data such as terrain and props. Access the editor by opening the game, clicking `Configure`, clicking the `Advanced` tab, clicking the box left to `Enable Edit Mode` so it is checked, clicking `Play`, going in-game to the scene you want to edit, pressing `ALT+ENTER`, and finally `CRTL+E` to enter and exit Edit Mode.

Sounds

Use Audacity for OGG, WAV, and MP3 files. To modulize, simply change scan_module_sounds = 0 to scan_module_sounds = 1 in module.ini if it isn't already. This folder contains audio specifically for sound effects.

Textures

For DDS, TGA, and JPEG files you can use GIMP or paint.net. To modulize, simply change scan_module_textures = 0 to scan_module_textures = 1 in module.ini if it isn't already. The folder contains image files used for different kinds of files used in materials such as diffuse textures, bump maps, environment maps, and specular maps. Diffuse textures are used to color materials. Bump maps look blue or green and provide some depth to your material with lighter areas being raised up and darker areas being sunken in. Environment maps are reflections that show up on your material. Speculars use a similar technique as bump maps but are black and white and determine how reflective a material is instead.

main.bmp

A BMP file for which you can use GIMP or paint.net. Simply localize to modulize, otherwise reads from the common main.bmp file. This is the image that appears on the splash screen when the mod is selected. Just remember before exporting, click the + to the left of Compatibility Options, then click on the box left of Do not write color space information to check it.

module.ini

Use Notepad for the INI file. This file is always local and modulized and contains various mod configuration settings such as the number of hints to display at num_hints and tells the game which BRFs to load.

map.txt

A TXT file which is always local and modulized. As of yet, there are no updated tools to easily edit this. Thorgrim's map editor may be old, but still works, Bloodpass Warband Map Editor would be an alternative.

*.txt

For the many other TXT files which aren't mentioned use the appropriate Module System. Always local and modulized. These files contain data on which files of every other type to use in the mod and pretty much all the inner workings of the mod.

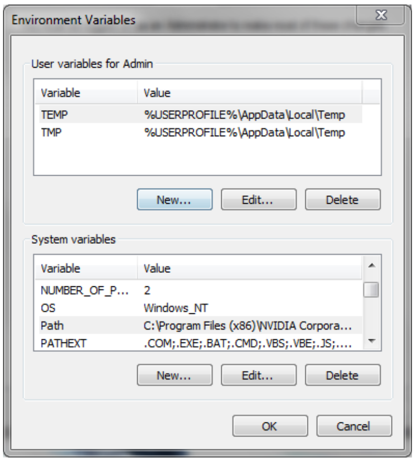
3 Setting up the Module System[3]

Are you finished with the theory? I want to start modding now.

Yes, I am. So, now let's start the modding. You want to setup a Module System, don't you? Well then, just go to the thread where we summarise downloads, the Warband Modder's Download Repository, and get yourself a fresh ModSys for your particular game of choice. (You're looking for the "Download Module System" section, can't miss it!)

Now. In order to be able to compile the files into game-readable text files, full of numbers and numbers and numbers, and to make your awesome mod come true, you need Python (the language). Having a real pet python might help, but don't count on it. Now, don't go rushing into downloading the latest Python. Here's something really important that you should know: The ModSys does NOT work with any Python newer than 2.x[4]. Fortunately, the same thread you've already got open also features a link to the Python website, so you've been saved a Googling; grab the latest 2.x.x version from there.

Now all you have to do is to install Python. Using Admin rights, please. Oh, it's a good idea to install it in its preferred C:\Python27 directory (if you're still using Python 2.7 in your day and age, oh reader from the future!). Next up. Go to your Environment Variables to see if everything is OK. For those that don't know what Environment Variables are - well, these are OS things, which your Windows uses. You will only have interest in the "PATH" variable. Now open that window and get to business.



User variables for Admin

Variable	Value
TEMP	%USERPROFILE%\AppData\Local\Temp
TMP	%USERPROFILE%\AppData\Local\Temp

System variables

Variable	Value
NUMBER_OF_P...	2
OS	Windows_NT
Path	C:\Program Files (x86)\NVIDIA Corpora...
PATHEXT	.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;....

Windows XP -> My Computer (right-click) -> Properties -> Advanced -> Environment Variables

[3] Taken from *The Ultimate Introduction to Modding* by Lumos.

[4] That's actually a lie. If you're familiar with Python, you're free to re-write the Python "compiler" that the ModSys uses, and thus convert it to Python 3.

Windows 7,8 -> My Computer (right-click) -> Properties -> Advanced system settings -> Environment Variables

Scroll that list until you find the "Path", then click it and hit the "Edit" button down there. This Path variable is basically just a string of paths to certain things which are important for your OS. It's different on every computer. In order for your compiler to work without any hacks or fixes, you must have your Python folder in your Path variable. For an example, if you did install your Python in C:\Python27, then you should add

```
;C:\Python27
```

at the end of the Path. Do **NOT** alter the Path in any other way! As it contains important system stuff, it's definitely **NOT** recommended. Just add that ;C:\Python27 if it's not there. I think it won't be there, since I had to enter it manually. I think.

Anyways, notice the semicolon (;) that is before the C:\Python27? Entries for different applications are separated with semicolons in the Path variable string, so you should add that semicolon so your OS knows what you're talking about. For Windows 10, be sure not to include the semicolon because it will cause an error during the module setup. And that's about it for setting Python up.

And where is the modding?

Modding comes next. Are you OK? Still alive, computer not exploded? Very well, keep reading! You've set your Pythonic compiler up, now it's time to compile the basis for your new mod.

Now, make sure you have permissions to read and write in your ModSys folder and in your Mount&Blade folder (best is if you can freely change the whole folder). Now, open up your Mount&Blade\Modules, copy Native and paste it in the same place. Change the folder name to your new mod's name.

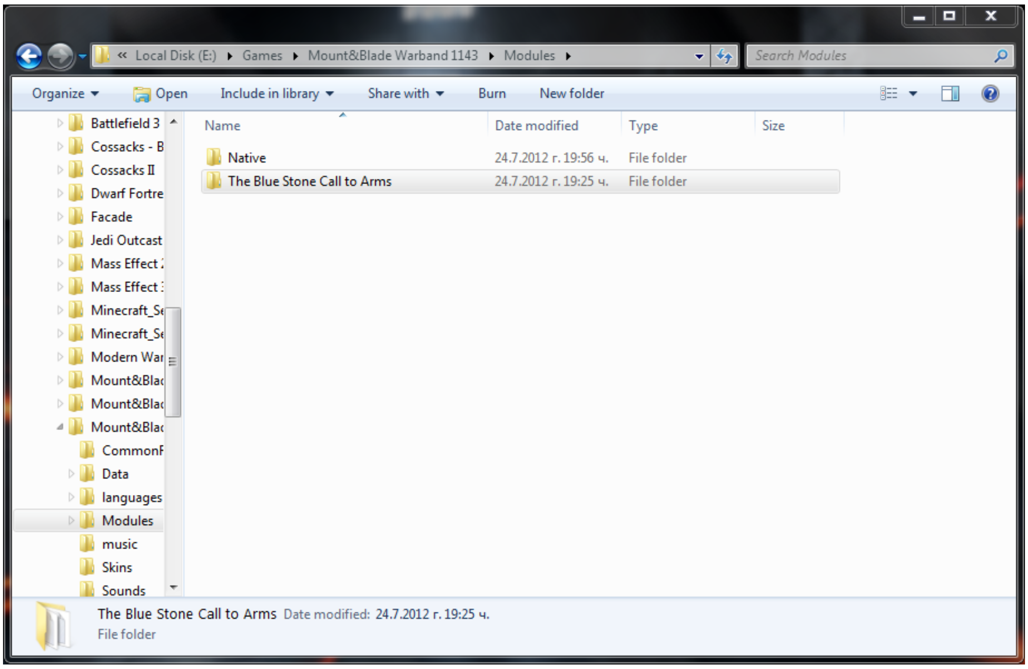
Now that you have your new mod folder, you have to make your ModSys use it. Open up your ModuleSystem folder, locate

```
module_info.py
```

and open it. **Important!** Do **NOT** double-click the file. If you do it, Python will think that you're trying to compile a Python program, and will basically do nothing. In order for you to edit ModSys files, you need to use a text editor. You can use Notepad, although I'd strongly recommend using Notepad++, a very powerful and free text editor. Python ships with IDLE, a text editor, but it's old and bad, don't use it. Anyways, open up your module_info with your favourite text editor, and you will see this:

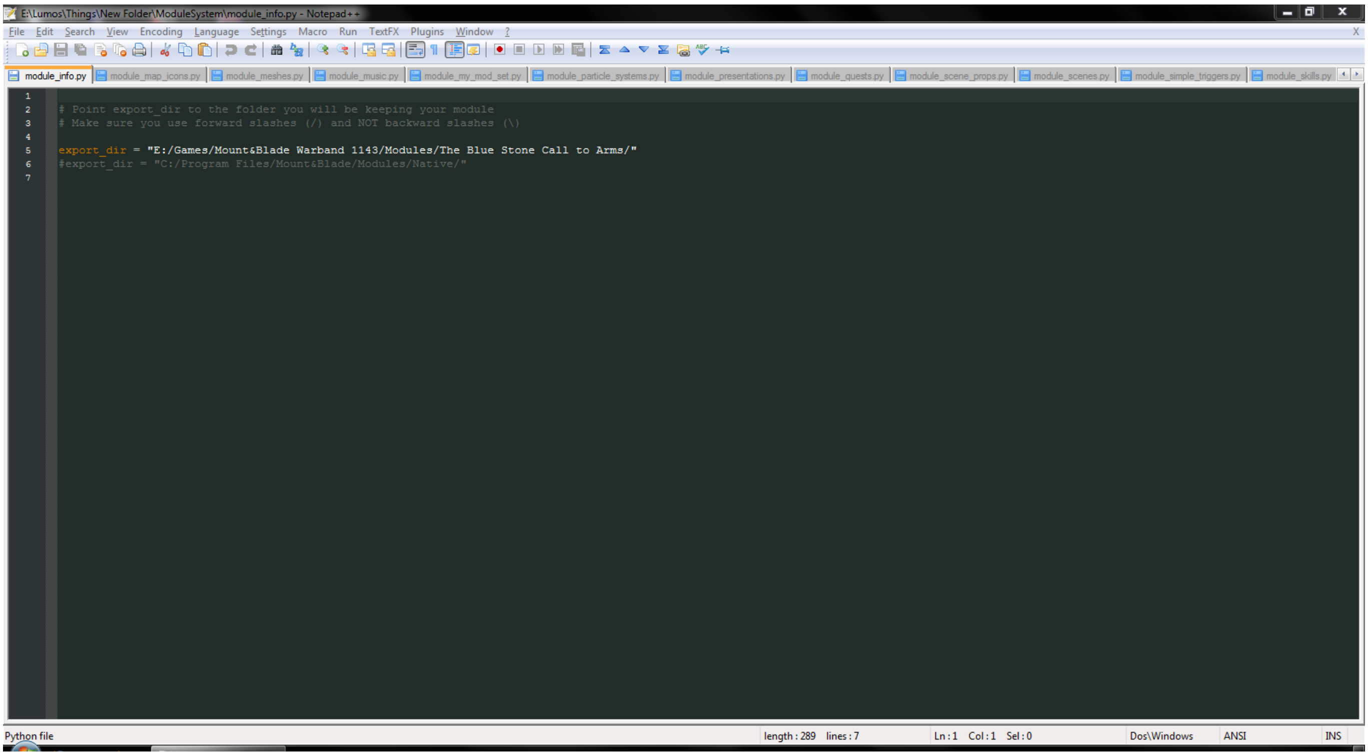
```
# Point export_dir to the folder you will be keeping your module
# Make sure you use forward slashes (/) and NOT backward slashes (\)

```



```
export_dir = "../WOTS/Modules/Native/"
#export_dir = "C:/Program Files/Mount&Blade/Modules/Native/"
```

Now this right here, “WOTS“, is a remnant from pre-2005 (or so), when the game was (temporarily) called “Way of the Sword“. If you’ve played the very first versions of M&B, you can see old stuff still hanging around.



```

# Point export_dir to the folder you will be keeping your module
# Make sure you use forward slashes (/) and NOT backward slashes (\)

export_dir = "E:/Games/Mount&Blade Warband 1143/Modules/The Blue Stone Call to Arms/"
#export_dir = "C:/Program Files/Mount&Blade/Modules/Native/"

```

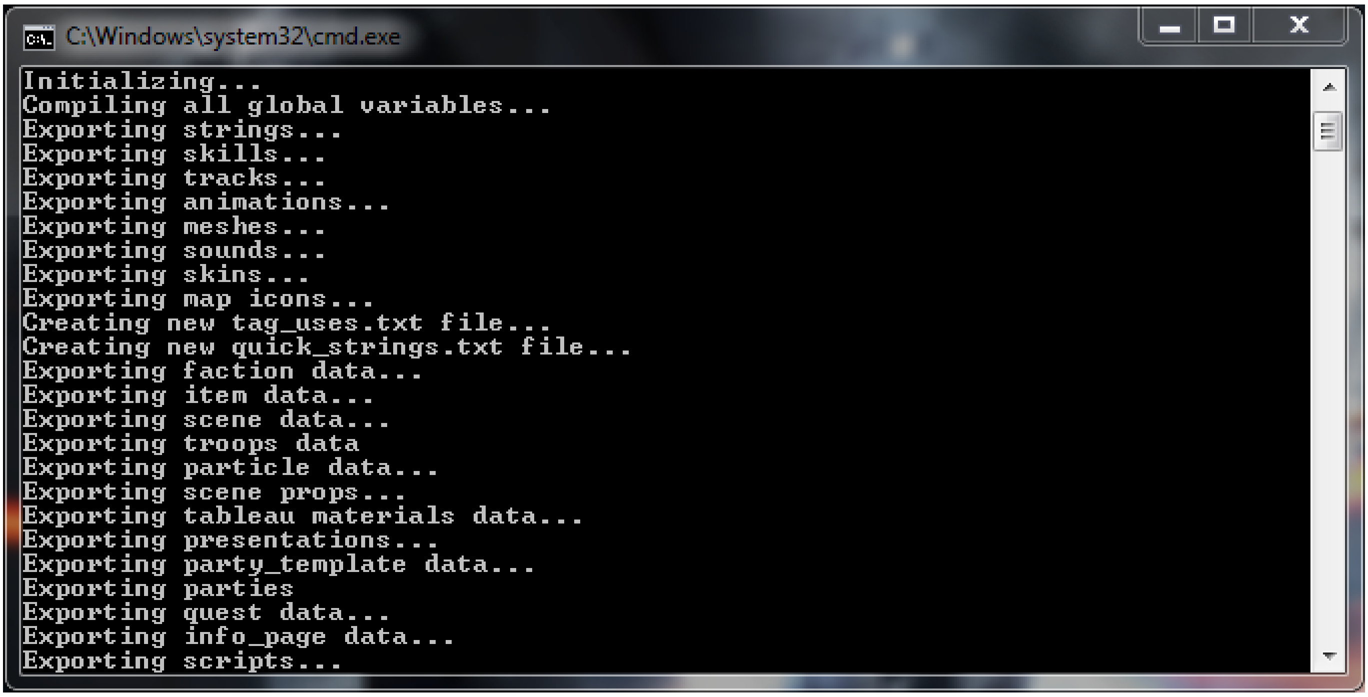
Right. So, you see what you have to do. Now, let’s give you some overhead. The # mark you see right there means that this line is a comment. If you’re a programmer, you know what a comment is, if you're not - a comment is something that stands there for the programmer, not for the computer. Comments are used to take notes, make documentation and whatnot. Damned useful, comments are. Everything followed by a comment is invisible to the compiler, so if someone tells you you have to change both lines, tell them to **** off.

Change your export_dir (the non-commented one) so that it points to your new mod folder, and follow the lead for the slashes.

As you can see, the second (commented) line is not changed, and it works OK. Now, you've set your mod up. And here comes the first challenge.

Challenge? What challenge?

Don't worry, it's nothing. You have to double-click the build_module.bat file in your ModSys folder - this is the compiler. Now a console window will pop up and you will see some output. If you've done everything correctly, you'll see a good output without any errors and you can be happy.



```
Initializing...
Compiling all global variables...
Exporting strings...
Exporting skills...
Exporting tracks...
Exporting animations...
Exporting meshes...
Exporting sounds...
Exporting skins...
Exporting map icons...
Creating new tag_uses.txt file...
Creating new quick_strings.txt file...
Exporting faction data...
Exporting item data...
Exporting scene data...
Exporting troops data
Exporting particle data...
Exporting scene props...
Exporting tableau materials data...
Exporting presentations...
Exporting party_template data...
Exporting parties
Exporting quest data...
Exporting info_page data...
Exporting scripts...
```

But if you've failed to do something from the above, you can get errors. The highest chance of an error you'll get is to see

```
'python' is not recognized as internal or external command, operable program or
   batch file.
```

about twenty times in a row. What this means is that you haven't set up your "Path" variable properly. Feel free to consult the stickied How to fix "'Python not recognised..."' thread in the forum. If/When everything works OK, keep reading.

Uh... I'm getting a "can't open file" or something" error! What do I do?

You must be sure to have Read/Write permissions to your ModSys folder, and to your new mod's folder, of course. You can try running the build_module.bat as an admin, it may work - I haven't tried it, I don't know. However, putting the ModSys folder on the Desktop or in My Documents is bound to fix the problem.

I'm getting another error!

Use the forum's search function (it's in the top-right corner) or ask a question in the Modding Q&A board. Do not make a new thread specifically for your question.

All right, my mod compiles properly. What now?

Well, my friend, this means you're done. You have the basis for your mod completed, and all that's left is the real modding itself. One more thing you should know, though. As you probably noticed, there are four types of .py files in the ModSys folder. Read on below to understand their purposes.

3.1 header_*files

Header files contain constants for the game. You can change them, but it's generally not recommended or needed. If you want to define yourself a new constant, you can do it in the appropriate module_* file. All operations are documented in header_operations.py.

3.2 ID_*files

ID_* files contain the numerical indexes for everything in your mod. In MABL/MBScript/whatever, everything can be looked at as a list of stuff represented by an index, and ID_* files contain these indexes. ID_* files get re-created with every compile you do, but don't change them. You can use ID_* files to check what indexes your stuff has - if you get game errors that report an index but not a name, for an example.

3.3 module_*files

As you already saw, we edited module_info.py. Module_* files contain all the game stuff, and they are the ones that you will edit. They contain nearly everything - troops, items, scene props, scenes, etc. You are allowed to edit them. Backing up your ModSys folder from time to time is recommended.

3.4 process_*files

The process_* files are responsible for compiling your mod into the game's beloved numbers. Modification to these files seldom happens, and before you do anything, you have to be absolutely sure that you know what you're doing - but for a start, just don't touch them. The process_* files, unlike the other parts of the ModSys, are written in real Python.

But wait, aren't you going to tell me how to mod?!

This is a path that everyone must walk on their own. In other words, no, I'm not going to tell you how to mod. I will, however, give you some tips and links that will be useful for you.

Fine. What are they?

Here we go...

- Firstly, grammar and attitude. Remember these two words, for they are very important. Write properly, be polite to others, and your chances of getting help will increase exponentially. Put spaces after punctuation marks, remember that "u" and "y" and other crap are not words but letters, and no "halp plox". Treat others as your equals, in the way that you want to be treated. Members with experience (that even spare time to answer your questions) should be treated with all the respect you (should) treat your mentors with. Don't be afraid to ask your questions and to use the word "please".

- Stickied topics are your friend. Stickied topics are threads on the forum which always stay on the top, and do that because they are important. Reading them is always a good idea.

- Read the forum rules. They aren't a lot, and are put quite shortly and understandably. Also remember to read the box that appears when you make a new topic. Read it at least once, it won't hurt you.

- If you have a small question, do not make a topic about it. Post your question in the Modding Q&A board instead.

- Use the forum search and/or Google search on the forum to find things you need.

- The best way to learn modding is to try. Start off small, although that may seem boring as an advice. But trust me, I know it from experience. Starting off small is the good way to start; trying to make a gigantic project from the beginning is a very, very bad idea. Trust me.

- You can read what each "operation" (i.e. ModSys function) does in header_operations.py.

Wait, how can I edit the World map?

Since you asked, here's something important. In modding, the term **map** refers only to the World Map, the one where you move around towns with your little guy. Multiplayer maps, called like that similarly to other MP games, are called **scenes** in the modding section. Know that, because it's highly annoying to come across a person who asks how to edit a map, while he's actually referring to a scene.

But let's get back on the question. To edit the World Map, you need a Map Editor application. There are a few in existence, but the most well-known of them all is Thorgrim's MapEditor. It was made for the old M&B, but it can work with Warband, no matter what people may be telling you. An alternative is Bloodpass' Warband Map Editor.

Demonwolf's tutorial How to make Campaign Maps is very good (I've followed it myself) and will get you started on the actual map editing.

Also, please note that the larger the map, the more cumbersome it is. Maps with loads of vertices (yes, the map is actually a 3D mesh), like more than 80000, will put a lot of pressure onto people's computers and should generally be avoided.

Is that it?

That's about it. Now it's time for you to venture into the big wide world of modding.

- *Lumos*

4 Module Syntax and Usage[5]

4.1 The Basics

A listing of all of the valid commands and operations that can be used in the module system is found in the file header_operations.py. It includes comments (following the # sign) that may give a hint as to each operations use, but also identify what arguments need to be included with each operation. Every operation used in the module system takes the following form:

1. It is enclosed in parentheses: (operation)

2. It is immediately followed by a comma: (operation),

3. Arguments within the operation parentheses are separated by a comma: (operation, argument, argument)

Additionally, the header_operations file assigns a numerical code to each of the operations listed there. For instance, call_script = 1. When the game gives errors, they are in the form of SCRIPT ERROR ON OPCODE ##:.... OPCODE number provided refers to the number assigned to the operation in header_operations. So, if the error was with OPCODE1, the line of code in question used the operation call_script.

Items contained within the various module files are indexed, that is assigned an ordinal numeric value based on the order they are found in the file. These indexes begin with 0 and count by 1 to each next value. Thus, troops, items, triggers, scenes, etc, once compiled into .txt files, are referred to by their index value. So, if you see an error code in “trigger 0“, that is the first trigger in the indicated file, and if the error is in line 4, you are looking for the fifth operation within that trigger.

4.1.1 Variables

Local variables - `":variable"` - variables declared with a : are local variables. They only exist in the current code block–enclosed by brackets [ ] –and cannot be referenced in other code blocks unless passed as script parameters. One should initialize them (to 0 or another suitable value) before they are used as they may contain a value already.[6] They are enclosed with quotations . Ex: ":agent_id"

Global variables - `"$variable"` - variables declared with a $ are global variables. They are available to be referenced in any part of the code, from any module_*, once they are defined. Their default

[5] Taken from *An Introduction to Module Syntax and Usage* by Caba‘drin.

[6] If you have a script that is called repeatedly (from a loop) the local variables within this script are NOT reset with every call due to the quick succession of the script calls. One must initialize them at the beginning of the script so they are re-set with every call. Caba‘drin, Modding Q&A.

value is 0. They are enclosed in quotations . Ex: "$g_talk_troop".[7]

Registers - `reg(#)` OR `reg#` - are variables used by the engine to store bits of information temporarily. They are global in scope, but may be accessed by a variety of different scripts for their duration and are then left for another bit of code to use them. In header_common, registers reg0 to reg65 are declared.[8]

Strings - `s#` - are specific 'registers' used to hold a string, rather than an integer. Strings not declared in module_strings (quick strings) begin with the @ symbol. In header_common strings s0 to s67 are declared.

Positions - `pos#` - Unlike the previous two "registers", positions store more than one piece of information; each position stores X, Y, Z coordinates as well as rotations on each X, Y, and Z axis. In header_common positions pos0 to pos64 are declared, with pos64 beginning an unspecified range of positions used by siege towers (belfries).

Constants - `constant` - are constants that are declared in module_constants, maintain the numerical value assigned there, and cannot be altered save through editing module_constants. They can be called from any module_* file and any code block. Constants are not enclosed with quotations.

Local and global variables, as well as registers, are declared with the

```
(assign , <variable\_name>, <variable\_value>),
```

operation. The variable value field can be another variable name. Strings and positions have their own unique operations, as defined in header_operations

Keep in mind that registers and variables can only be used where they are meant to go: in conditions and operations blocks.[9]

[7] Global variables values are saved with saved games and a global variable declared in the start game script won't be redeclared when a game is loaded. Caba'drin, Modding Q&A.

[8] `reg(#)` is the old way which is still working but got replaced by `reg#` on newer versions of the Module System. You can can still see some leftovers on the Native Module System that were not updated; kalarhan, Modding Q&A.

[9] Yoshiboy, Modding Q&A.

4.1.2 Module System Prefixes

Within the Module System (eg module_mission_templates), one can refer to code or information stored in other files of the Module System by using a prefix before the label for the piece of code. Only module_dialogs.py has no prefix since it is never directly referenced.

module_animations:	"anim_"	module_pstfx:	"pfx_"
module_factions:	"fac_"	module_presentations:	"prsnt_"
module_info_pages:	"ip_"	module_quests:	"qst_"
module_items:	"itm_"	module_scene_props:	"spr_"
module_game_menus:	"menu_"	module_scenes:	"scn_"
module_map_icons:	"icon_"	module_scripts:	"script_"
module_meshes:	"mesh_"	module_skills:	"skl_"
module_mission_templates:	"mst_"	module_sounds:	"snd_"
module_music:	"track_"	module_strings:	"str_"
module_particle_systems:	"psys_"	module_tableau_materials:	"tableau_"
module_parties:	"p_"	module_troops:	"trp_"

4.1.3 Module_Scripts, Parameters, Arguments, etc.

Most of the module_* files contain self-contained code for the most part. Files such as _skills, _troops, _items etc. simply define these various elements for the game. module_game_menus contains all of the various menus, their options etc. Presentations has all of the more complicated presentation-drawing data, etc.

But module_scripts is referenced from many of these files–it contains “common“ functions to be called upon by every part of the game code. You can consider (call_script, "script_scriptname", <optional_aruments>), a Goto:, Jump, function call, what have you.

To fascilitate this process, the call script operation allows information to be passed from the current code to the script without using global variables. These arguments or parameters can be local variables generated by the current bit of code. The script code in module_scripts will then begin by storing the parameters passed to it into local variables for use throughout the script. Scripts cannot pass information back to the code they were called from in the same way though. To do this, registers are typically used. In the calling code, once the script has completed, these registers are recorded to local variables for ease of use. Up to 15 script parameters can be passed with any one call.

```
//...statements...//
(assign, ":local_variable_1", 10),
(assign, ":local_variable_2", 5),
(call_script, "script_example_script", ":local_variable_1", ":local_variable_2"),
(assign, ":local_variable_3", reg0), #local_variable_3 = 15
//...further stuff, working with local_variable_3...//
```

```
# script_example_script
 # Input: variable_1, variable_2
 # Output: Sum in reg0
 ("example_script", [
    (store_script_param, ":num1", 1), #now num1 is 10
    (store_script_param, ":num2", 2), #now num2 is 5
    (store_add, ":sum", ":num1", ":num2"),
    (assign, reg0, ":sum"),
 ]),
```

4.2 Terminology

4.2.1 Troop vs Agent

Troops refer to the entries in module_troops by the name assigned to them there. "trp_player" is the player, "trp_npc##" for the various companions, "trp_knight_#_##" for lords, and "trp_swadian_footman " etc, etc. With more generic soldiers, the entry in module_troops is used as a “template“ to create multiple instances of them in each party, etc.

Alternatively, agents refer to the specific actors, by number, in a scene (whether that scene is a battle, a tavern or what have you). Agents are created from the templates in module_troops, for the player, companions, NPC lords. They are still unique but with generic soldiers multiple agents can be created from a single troop template. Each agent has a unique number for that scene but when the scene ends the agent doesn't exist any more so its agent id is meaningless.

4.2.2 Parties and Party Templates

A “party“ is any entity on the world map, be that the player's party, bandits and their lairs, NPC lords, or a town/castle/village. The different types of parties are separated into categories in the first party slot "slot_party_type" (or party slot 0–see the discussion of slots below), which contains an integer associated with a certain type. These types are defined in module_constants and begin with the heading spt_* (for slot_party_type). Some examples of spt_* values are spt_kingdom_hero_party (for NPC lords) and spt_castle (...for castles).

Each party has an ID number that it is referenced by in the code. For instance, the player's party is "p_main_party" which has always an index value or Party ID of 0. Certain parties have constant or static ID numbers, such as the player's party and all towns, castles and villages. Those parties with static IDs are defined in module_parties. You can see their ID number in the file ID_parties.py after you compile your module.

For all parties NOT defined in module_parties.py, those are created, dynamically, by the game based on a party template, in a manner somewhat parallel to the agent/troop division above. These party templates are defined in module_party_templates and have names that are referenced with the prefix "pt_#". The template contains, primarily, a list of troops that are contained in that template and a numeric range that is used to determine how many of each troop type will be given to that template in game (the engine randomly picks a number within that range). A party template can be used to create a new party with the command <spawn_party>. This new party is assigned an ID number and then can be classified with a slot_party_type value, etc. Party templates can also be used with already existing parties. One can add a template (the troops defined in that template) to another party to “reinforce“ an existing party in this way.

4.2.3 Slots[10]

Slots are essentially ways to have variables arrayed by an object (party, troop, agent, team, faction, scene, item, player, scene prop, or party template). Each party, troop, agent, item etc has a number of “slots“ (variables) assigned to it. That variable has a common name "slot_item_SLOTNAME" for each version of the object is assigned to (items in this case), which is useful, but stores unique information for each object. Where it looks like a “constant“ is the fact that the engine turns the name of the arrayed variable (the slot) into a number in module_constants (slot_item_base_price = 53). This means that the engine looks at the 53rd slot-variable associated with a given item to find that item's base price.

So, when you see operations using slots such as

```
(item_get_slot, ":base_price", ":item_id", slot_item_base_price),
```

you can see that you need to specify exactly which item you want to get the base price for. But, since the variable is arrayed, you could embed that in a try_for_* loop and iterate through many ":item_id"s to do things to the base price of every item, say.

You can access the price for a specific item with the item_get_slot operation, change it via item_set_slot, and test for equality with item_slot_eq and the like.

If you know C-like programming languages, slots work a lot like this:

```
slot_troop_spouse
spouse[troop1] = spousename
spouse[troop2] = spousename
spouse[trp_player] = spousename
```

[10] With additional information bits by Dalion.

which the engine reads as troop_variable_30[troop1], troop_variable30[trp_player] etc. since the slot slot_troop_spouse is set as slot 30 in constants.

```
slot_item_base_price
baseprice[item1] = priceA
baseprice[item2] = priceB
baseprice[item3] = priceC
```

and the engine reads that as item_variable_53[item1], item_variable_53[item2] etc. since it is set as 53 in constants.

Before being set with a *_set_slot operation, the value of all slots is 0 like for a global variable[11]. Each slot has unique storage for a large number; each slot number can store a 64 bit integer, but other parts of the game engine (like multiplayer network messages) can only deal with 32 bit integers, so you should keep within the -2147483648 to 2147483647 range to be safe. For each type of object you can use slot numbers (the ones defined in module_constants.py) from 0 to 1048575 (2^20), but you should keep the slot numbers fairly close together to avoid wasted memory usage - the engine allocates a continuous array of slots up to the highest number used.[12] If you try to use slot higher than the upper bound, it won't be assigned, and retrieving its value will always give 0.[13]

Slots are globals tied to some object.[14] For instance, troop slots define what lords personalities are since they are unique for each lord. Agent slots define which agents are running away since it's also unique behavior for each agent and using global variables for this just won't do the job. Once you understand how slots work, you will always choose them over globals in some situations, not only because there is no other way, but sometimes because the feature will work better this way. Three things to remember when using slots:

1. They can be defined in any file the are used in, not just module_constants. Make sure though that other files in which this slot is in have the file with definition included on the top with *. Or they could not be defined at all: if you are sure that you will remember their meaning you can use (agent_set_slot, ":agent", <slot_no>, <value>) instead of (agent_set_slot , ":agent", <slot_name>, <value>). This is actually true for any constant, not just slots.
2. You can have multiple slots with the same number but make sure they are different "category". I.e. it is fine to have troop slot 145 and party slot 145 but avoid using two troop slots with number 145.

[11] Which is because global variables are slots as cmpxchg8b remarks in the Modding Q&A thread.
[12] Vornne, Modding Q&A.
[13] Dalion, Mount & Blade Modding Discord.
[14] Whole passage by Dalion, Mount & Blade Modding Discord.

3. Slot names are not saved anywhere during compilation, so they can be used to make your code more difficult to understand in compiled form (if that is what's you are after). When someone looks on code of your compiled module, to understand a global sometimes it's enough to look at its name. But to understand a compiled slot, you should look up all its uses and make conclusions based on them so it's far more difficult.

Should you face slots with appearantly wrong naming, like `slto_kingdom_hero` for example, take note that slto stands for `slot_troop_occupation`, marking the constant values for these slots. So it is not a spelling mistake which needs to be corrected, merely a naming convention of TaleWorlds.[15]

Not sure how to work in phantom values in slots[16]. There are also different comments with varying values[17] and some most probably outdated informations which might be interesting **if** still valid.[18] Comment of cmpxchg8b about game engine dealing with slots[19]

[15] Somebody, Modding Q&A.

[16] MadVader, Modding Q&A.

[17] _Sebastian_, Modding Q&A and Modding Q&A.

[18] Hellequin, Modding Q&A and Modding Q&A.

[19] cmpxchg8b, Modding Q&A.

4.3 Control and Conditional Operations and Syntax

In general, it is imperitive to note that the game engine treats every conditional operation as an “IF“ and all of the code that follows after a conditional operation as the “THEN“ in an If-Then statement. As the engine goes through lines of code, it will stop any time a conditional operation fails, and it will not read the remainder of the code.

To isolate If-Then(-Else) statements to allow failure but continue processing the remainder of the code, one must use a “try block“ with (try_begin), (else_try), and (try_end), This is discussed in more detail below.

4.3.1 List of Conditional Operations

```
(eq,<value>,<value>),   Does Value 1 = Value 2?
(gt,<value>,<value>),   Is Value 1 > Value 2?
(ge,<value>,<value>),   Is Value 1 >= Value 2?
(lt,<value>,<value>),   Is Value 1 < Value 2?
(le,<value>,<value>),   Is Value 1 <= Value 2?
(neq,<value>,<value>), Does Value 1 != Value 2?
(is_between,<value>,<lower_bound>,<upper_bound>), Is Lower <= Value < Upper?
```

Also, any operation that includes "_is_" can be considered a true-false condition test. For instance, (agent_is_alive,<agent_id>), will do a conditional test to check if the agent ID provided is alive. If the test is true, the code continues. If it is false, the code stops.

To test for “FALSE“ rather than “TRUE“ in any "_is_" T/F conditional test, use the following:

```
(neg|<operation>),
```

From our previous example,

```
(neg|agent_is_alive,<agent_id>),
```

the code will only continue if the agent identified by the ID number is dead (NOT alive).

4.3.2 List of Control Operations

```
(try_begin),
(try_end),
(else_try),
(try_for_range,<destination>,<lower_bound>,<upper_bound>),
(try_for_range_backwards,<destination>,<lower_bound>,<upper_bound>),
(try_for_parties,<destination>),
(try_for_agents,<destination>),
```

Destination is the variable that will be iterated in a loop. The iteration will begin at the lower bound and go to Upper Bound -1

4.3.3 Boolean AND and OR

Since, as discussed above, the engine treats every conditional operation as the “IF“ in an If-Then and all the code beneath it as the “THEN“, the boolean AND can be accomplished simply by stringing multiple conditional operations after one another.

For instance, if one wanted to locate an agent who is alive AND a horse AND an enemy, the code would simply be:

```
(agent_is_alive,<agent_id>),
(neg|agent_is_human,<agent_id>),
(neg|agent_is_ally,<agent_id>),
```

To check for one condition OR another, the following is used:

```
(this_or_next|<operation>),
```

By our previous logic, it can be strung together multiple times as well. For instance, to see if a variable is 1 OR 5 and, if so, continue:

```
(this_or_next|eq, ":test_variable", 1),
(eq, ":test_variable", 5),
```

4.3.4 If-Then-Else

Since all condition operations apply to every line of code beneath them, there needs to be a way to separate portions of code so the condition operation only applies to one section and then the code continues on regardless of the result of the test.

To do this, one uses a “try block“, code enclosed by (try_begin) and (try_end). One can think of them as creating a typical If-Then statement, with the first lines of code after the (try_begin) being the “If“ in the form of conditions tests and the following lines being the “Then“ with consequence operations, which are closed by the “End If“ of (try_end)

As in all good If-Thens, (else_try) emerges as the “Else If“ statement. At any point that a condition fails in the try before an (else_try) the engine will begin looking in the subsequent (else_try). Example:

```
#Code above, doing whatever
(try_begin),
  (eq, 1, 1), #True, go to next line
  (gt, 2, 5), #False, go to else try
  #consequence operations here
(else_try),
  (eq, 0, 1), #False, go to next else try
  #consequence operations here
(else_try),
  (eq, ":favorite_lance", ":jousting"), # Maybe?
        (assign, ":prisoners", 100),
(try_end),
#Code continuing on, regardless of whether the favorite lance=jousting and prisoners was set to 100
```

4.3.5 Conjunctive and Disjunctive Normal Form[20]

Stringing multiple conditional operations after one another can sometimes lead to quite complex checks which evaluate eventually in a conjunctive normal form (CNF) or a disjunctive normal form (DNF). A CNF is for those who know mathematical logic a conjunction of disjunctions, a stringing like if (a or x) and (b or y) and (c or z). Such one can easily be expressed in MABL as follows:

```
(this_or_next|eq,"$variable_a",1),(eq,"$variable_x",1),
(this_or_next|eq,"$variable_b",1),(eq,"$variable_y",1),
(this_or_next|eq,"$variable_c",1),(eq,"$variable_z",1),
```

If you want to express a stringing of conditional operations like if (a and x) or (b and y) or (c and z) then you want to express a DNF. There are two different ways to achieve such via MABL. The more intuitive way, depending on your familiarity with logical expressions, is longer and uses more registers as can be seen here:

```
(try_begin),
  (eq,"$variable_a",1),(eq,"$variable_x",1),
  (assign,reg(1),1),
(try_end),
(try_begin),
  (eq,"$variable_b",1),(eq,"$variable_y",1),
  (assign,reg(2),1),
(try_end),
(try_begin),
  (eq,"$variable_c",1),(eq,"$variable_z",1),
  (assign,reg(3),1),
(try_end),
```

[20] Summary of three notes by fisheye and Winter, Modding Q&A.

```
  (this_or_next|eq,reg(1),1),
  (this_or_next|eq,reg(2),1),
               (eq,reg(3),1),
 <block continues with whatever you want>
```

The alternative approach would be to convert the DNF to a negation of a CNF using the De Morgan's laws. After the conversion the above wanted string of conditional operations would thus be if ~((~a or ~x) and (~b or ~y) and (~c or ~z)) then. The negated CNF is then expressible in MABL as follows:

```
(try_begin),
  (this_or_next|neq, "$a",1),(neq, "$x",1),
  (this_or_next|neq, "$b",1),(neq, "$y",1),
  (this_or_next|neq, "$c",1),(neq, "$z",1),
(else_try),
  (do stuff here),
(try_end),
```

It might not be immediately obvious to you that the negated CNF is in fact equivalent to the DNF you want. Just take your time to think about it.

4.3.6 For-Next Loop

The M&B Module System has a number of different variations of the For-Next loop. The most basic is accomplished with a (try_for_range, ":iterator", <lower bound>, <upper bound>), then looped operations and capped by a (try_end), instead of a "Next iterator" or what have you.

The iteration will begin at the lower bound and go to upper bound -1, making single integer steps, counted by the variable assigned as destination, in the example above the local variable ":iterator". The ":iterator" must be used in the code that follows in the try_for_range block, but it cannot be altered to skip integer-step iterations toward upper bound -1. If the ":iterator" is not used in the code that follows the variable ":unused_iterator" MUST be used in its place.[21] Otherwise loads of warnings about the unused iterator variable will come up.

Example:

```
(try_for_range, ":i", 0, 10),
  (store_add, ":count", ":i", 1),
  (assign, reg0, ":count"),
  (display_message, "@{reg0}"),
(try_end),
#This code will display a message on screen counting from 1 to 10.
```

[21] Credit to Masterancza/Garedyr for clarification about ":unused_iterator", Mount & Blade Modding Discord.

There is also (try_for_range_backwards, ":iterator", <lower bound>, <upper bound>), where the iteration will begin at upper bound -1 and count down to lower bound. Ignore the comments in header_operations that say to switch the order of the lower and upper bounds. They are wrong. This is useful for removing something from a list (members from a party, for instance) without messing up the indexing of that list.

Note, that both the lower bound and the upper bound need not be a “plain number“; they can easily be variables that are set earlier in the code. Examples of this in the next section.

Two more specific (try_for_*) loops exist: (try_for_agents, <agent_id>), and (try_for_parties, < party_id>),. These will cycle through all agents in a scene or all parties in the game, respectively. The iterator represents the Agent's (or Party's) ID number and should be stored in a local variable. Example:

```
(get_player_agent_no, ":player"),
(agent_get_team, ":playerteam", ":player"),
(try_for_agents, ":agent"), #Loops through all agents
  (agent_is_alive, ":agent"), #Checks if current agent is alive
  (agent_is_human, ":agent"), #If alive, checks if current agent is human (not a horse)
  (agent_is_non_player, ":agent"), #If alive and human, checks that the current agent isn't the player
  (agent_get_team, ":team", ":agent"), #If alive, human and non-player, gets the current agent's team.
  (eq, ":team", ":playerteam"), #Checks that the current alive, human, non-player agent is on the player's team.
      #Consequence operations go here to do stuff with this alive, human, non player, ally agent.
(try_end), #Go to next agent
```

4.3.7 Loop Breaks

Many times you may want to loop through all agents to locate one particular agent, or loop through another range of things (items for instance) to locate one particular weapon, and there is no need to loop through the rest–you've already found what you needed. Enter the loop break. The M&B Module System does not have any operations specifically to accomplish this, but there are a few simple tricks to effectively break a loop. The trick to use depends on the type of Module System (try_for_*) loop you are using.

4.3.7.1 Method 1: Changing the Loop's End

- *Breaking a try_for_range loop:*

You can break a try_for_range loop easily by changing the upper bound so it equals the lower bound. That way, when the engine tries to iterate to the next instance of the loop, it appears the loop has already reached the last iteration and is complete. The loop will exit without running the code again. To do this, the loop's upper bound must be a variable defined before the loop begins, and it must be variable that can be modified without loss of data–make a new variable or a copy of another if necessary. Once the code has been executed enough times, something has been found, etc, set the variable for the end of the loop to equal the loop's lower bound (either assign it the variable value or the constant value that the loop began with). Example:

```
#Within a larger (try_for_agents) loop
  (assign, ":end", "itm_glaive"), #Set the loop's upper bound
  (try_for_range, ":item", "itm_jousting_lance",":end"), #Loop through the lances
      in the game
    (agent_has_item_equipped, ":agent", ":item"), #Check if the agent has the
        current lance equipped
    (agent_set_wielded_item, ":agent", ":item"), #If the current lance is equipped
        , make the current agent wield it
    (assign, ":end", "itm_jousting_lance"), #loop breaker - make the upper bound
        equal the lower bound - stop looking through the lances
  (try_end),
#Continue on within the try_for_agents loop
```

- *Breaking a try_for_range_backwards loop:*

The same method is used for a backwards loop. This time, however, the lower bound is the "loop's end" so the lower bound will need to be the variable changed, and it should be changed to equal the upper bound. Example:

```
  #Code above...sets the variable ":troop" to some value
  (assign, ":array_begin", 0), #Set the loop's lower bound
  (try_for_range_backwards, ":i", ":array_begin", 10), #Loop from 0 to 10
    (party_slot_eq, "p_main_party_backup", ":i", 0), #Check if the ith party slot
        for the player's party backup equals 0
    (party_set_slot, "p_main_party_backup", ":i", ":troop"), #If it does, set that
         slot to the variable troop's value
    (assign, ":array_begin", 10), #Loop breaker - make lower bound equal the upper
         bound - don't check other slots
  (try_end),
```

4.3.7.2 Method 2: Using an Condition Test

- *Breaking a try_for_agents or _parties loop*

With try_for_agents and try_for_parties loops, it is not possible to change the loop's "upper bound". Instead, a conditional test (typically for equality) is used at the start of the code to execute within the loop. So long as it is true, the code within the loop will run. Once it is set false, the loop will keep going, but nothing inside the loop will happen since the first condition always fails - the loop will end quickly.

To do this, set a variable up before the loop with a given value (it is easiest to create a new variable with the value 0). Then, just inside the loop, set up an conditional operation testing for that value. After the block of code you want only ran once, change the value of the variable so the condition test fails the next time the loop runs.

Example:

```
#Code above...sets pos1 to some value
  (assign, ":break_loop", 0), #Set up variable to break loop and a value to test
  (try_for_agents, ":agent"), #Loop through all agents
    (eq, ":break_loop", 0), #See if the loop-breaker is still true
    (agent_is_alive, ":agent"), #If so, keep going; check if the current agent is alive
    (agent_is_non_player, ":agent"), #If alive, check it isn't the player
    (agent_is_human, ":agent"), #If alive and not the player, check that is is human
    (agent_get_position, pos0, ":agent"), #If alive, human and non-player, get it's location and record to pos0
    (get_distance_between_positions_in_meters, ":distance_to_target_pos", pos0, pos1), #See how far pos0 is from pos1 and record that in the local variable
    (lt, ":distance_to_target_pos", 10), #Check if the alive, human non-player agent is within 10 meters of pos1
    (assign, ":agent_at_target", ":agent"), #If so, record this current agent's ID to the local variable "agent_at_target"
    (assign, ":break_loop", 1), #Agent Loop Breaker - change the test variable so the equality test fails on the next try
  (try_end),
#The loop will break the first time an alive, human, non-player agent is found within 10 meters of the target position pos1
#Further code can now do stuff with the agent that was found
```

4.3.8 Loop Extension

In a similar way like you can break a loop if a specific conditions is met you can also extend a loop if a specific condition is (not) met. See as an example the following script:

```
(store_add,":range_end",towns_begin,1), # ":range_end" = towns_begin+1
(try_for_range,":town",towns_begin,":range_end"), # try for towns_begin to
    towns_begin+1
    (eq,":town",":found"), # if condition is true . . .
    # more here
(else_try),
    (neq,":town",":found"), # if condition is not true . . .
    (val_add,":range_end",1), # ":range_end" = ":range_end"+1, i.e. continue the
        try_for_range
(try_end),
```

This loop basically keeps adding 1 to the upper bound of the `try_for_range` if the conditions aren't met.[22]

4.4 Intendation

Write some lines about it here. Declarations outside body can't be tabbed, Python intendation matters and VC auto intendation[23]

[22] Winter, M&B Scripting Q&A.

[23] kalarhan, Modding Q&A.

5 Documentation Module System[24]

5.1 Introduction, Configuration and Interaction

The module system uses Python lists to represent collections of game objects. A Python lists starts with a square bracket [, includes a list of objects and ends with a closing square bracket]. If you open and view any of the module files you will see that it usually contains such a list. We call the objects tuples. Tuples, like lists, contain elements seperated by commas but they start and end with parentheses. Take note that there is no need for a comma when there is only one entry in a list, although the comma is needed when there is only one entry in a tuple.[25] The structure of each tuple object is documented at the beginning of the respective module file.

In this guide you will find this documentation always at the beginning of each module_*.py subchapter. Examples for module_*.py files which differ from the normal structure are module_info.py, module_info_pages.py and module_constants.py which will also get addressed first. After observations and examinations of example tuples there will also be all available flags listed which can get used by you. **Be aware:** All flags with mask and bits in the name cannot and should not be changed by you! They will therefore also not be listed.

5.1.1 Module Info

```
# Point export_dir to the folder you will be keeping your module
# Make sure you use forward slashes (/) and NOT backward slashes (\)

export_dir = "../WOTS/Modules/Native/"
#export_dir = "C:/Program Files (x86)/Mount&Blade Warband/Modules/Native/"
```

module_info is only needed for setting up the Module System as described in Chapter 3.

5.1.2 Interaction between Module System Files

[24] Partially taken from *Updated Module System Tutorial* by jik and for some clear descriptions from the Mount&Blade Wiki Fandom.

[25] cmpxchg8b, Modding Q&A.

5.2 Gameplay-related Module System Files

5.2.1 Module Constants

```
These constants are used in various files.
If you need to define a value that will be used in those files, just define it
    here rather than copying it across each file, so that it will be easy to change
     it if you need to.
```

module_constants.py is a very simple file: It is filled with constants – you should already know a bit about them from reading the Subchapter 4.1.1 of this Modding Guide. The constants defined here work exactly the same way as constants defined anywhere else. However, constants which are used in multiple module files should be defined in module_constants so that the module system always knows where to find them. It also serves as an organised, easily-accessible list where you can change the value of a constant whenever you might need to. Any changes will immediately affect all operations and scripts using the constant, so you don't have to manually find and replace them. For example, changing

```
hero_escape_after_defeat_chance = 80
```

to hero_escape_after_defeat_chance = 50 would immediately change any operations or scripts using hero_escape_after_defeat_chance, treating the constant as the value 50 instead of the value 80.

This file also stores the range markers such as kings_begin = "trp_kingdom_1_lord" and kings_end = "trp_knight_1_1". It is recommended to first look up these ranges and to get a grasp of them since, to stay with the example of troops, it determines at which places you should implement new kings, merchants, mercenaries, etc. in module_troops.

Another powerful tool defined in the constants section is the use of slots about which you should have read already in Subchapter 4.2.3. Slots can be assigned to any tuple object (troops, factions, parties, quest, etc.) and can store a specific value that is used by all in that object type. These are used as indices by the Module System for object_get/set_slot to fetch and/or store data in array format for the various objects that exist in-game.[26] Let's look at where the definition for quest slots are listed as an example:

```
###########################################
## QUEST SLOTS
###########################################
slot_quest_target_center = 1
slot_quest_target_troop = 2
slot_quest_target_faction = 3
slot_quest_object_troop = 4
##slot_quest_target_troop_is_prisoner = 5
slot_quest_giver_troop = 6
slot_quest_object_center = 7
slot_quest_target_party = 8
slot_quest_target_party_template = 9
slot_quest_target_amount = 10
slot_quest_current_state = 11
slot_quest_giver_center = 12
slot_quest_target_dna = 13
slot_quest_target_item = 14
slot_quest_object_faction = 15
slot_quest_convince_value = 19
slot_quest_importance = 20
slot_quest_xp_reward = 21
slot_quest_gold_reward = 22
slot_quest_expiration_days = 23
slot_quest_dont_give_again_period = 24
slot_quest_dont_give_again_remaining_days = 25

###########################################
```

[26] Somebody, Modding Q&A.

On the left side of the “=“ are the names given to each slot. The slot number is on the right side of the “=“. The names make it easier to know what the slot is used for. To be specific, "slot_quest_xp_reward" is used by native code to store how much XP you get from completing the quest. When referencing this value, you can either reference it as slot_quest_xp_reward or by the number 21. When looking through the code it’s easier to understand the name over the number.

If you want to add slots, simply define your new slot to be any value - if it’s too high, then objects will reserve memory up to that index and there will be a performance issues. If you define it to be a value that already exists, then the game will simply use the same space for both - it doesn’t really care what name you give it. Is your item for example a gun, you can safely use the same slot as food or books - item slots are rarely used in Native anyways. After you’ve done so, simply use (item_set_slot , "itm_gun", slot_item_whatever, val) from anywhere to store a value. This is always a number but that number can be a reference to another object - for example "snd_lyhyt" from module_sounds.py. Notice that the reference without quotes takes on the index value generated in id_sounds.py, while the quoted reference has probably the same value bitshifted to prevent conflicts.[27]

Make sure that the files at which you use the here implemented constants are referencing to module_constants.py via from module_constants import * at the top of the file, otherwise you might encounter an error at compiling.

[27] Somebody, Modding Q&A.

5.2.2 Module Variables

Write here a few lines about how useless this file is.

5.2.3 Module Strings

module_strings.py is arguably the simplest file in the module system. It contains strings, so blocks of text which can be displayed in various ways. Strings are used all throughout the module system, from module_dialogs to module_quests; whenever a block of text needs to be displayed on the screen. module_strings.py, however, is a repository for independent strings, which are not bound to any single file. Therefore they can be called by operations from anywhere in the module system. You can see these strings used in various places in the game, as white scrolling messages at the lower left of your screen, or in a tutorial box, or even inserted into a game menu or dialogue sequence.

Much like module_factions, this file immediately begins with a Python list: `strings = [`. Again, the first few tuples are hardwired into the game and should not be edited. Example of a string:

```
("bandits_eliminated_by_another", "The troublesome bandits have been eliminated by
    another party."),
```

or

```
("s5_s_party", "{s5}'s Party"),
```

Tuple breakdown:

```
1) String id. Used for referencing strings in other files.
2) String text.
```

One notable feature of strings is the ability to put a register value or even another string inside of it. You can do this by adding, for example, reg0 in the string. This will display the current value of reg(0). reg10 would display the current value of reg(10), and so on.

For additional strings, you must use the string register instead of a normal register; this is because strings are stored separately from registers. For example, s2 would display the contents of string register 2 and not the contents of reg(2). You can freely use string register 2 and reg(2) at the same time for different things, they will not overlap or interfere with each other.

String registers have two uses, and two uses only[28]:

1. You can put values in them using the various str_store_xxx commands (which you can find in header_operations.py); and you can clear them using the str_clear command.

2. They can be referenced in strings displayed using the display_message command, using the notation s# where '#' is the number of the string-register.

For instance, if you have an entry in your module_strings.py file that looks like

[28] Neophyte, Modding Q&A.

```
("lets_meet_in", "Let's meet in {s10}."),
```

and a script which contains the lines:

```
(str_store_party_name, 10, "p_zendar"),
(display_message,"str_lets_meet_in"),
```

then that piece of code will display the message

```
Let's meet in Zendar.
```

when it's executed. The s10 part in the string-entry means “put the current contents of string-register 10 here“, while the (str_store_party_name, 10, "p_zendar") part means “store the name of the party p_zendar in string-register 10“.

5.2.3.1 Strings and Syntax[29]

At strings you have to keep in mind some special Syntax rules. You might notice the usage of a \ at some places. It breaks the length of the string but does not add a carrage return in game, it just makes the code easier to read. A ^ is a nextline character, allowing you to make line breaks at the text in-game. A \ on the other hand is getting used as a Python trick to break the text into multiple lines (easier for us humans to read). You can not add any extra spaces on the start of the lines though or it will give a syntax error.[30] A {!} is getting used as a placeholder for an empty string, which otherwise causes problems for the game. It's usually followed by registers or strings, which can be null and won't display properly. It also excludes them from generating a line when you create translation .csv files.[31] It won't show up in-game unless it's a part of a string literal (i.e. party name). Also a "_" can get used inside a string which is treated like an empty space (the same is true at the names of items, troops, etc.).[32] E.g. the two strings ("cant_use_inventory_now","Can't access inventory now."), and ("cant_use_inventory_now","Can't_access_inventory_now."),, would display the same result in-game.

{reg#} will output the value of that register when used in a string. Using this method the following lines show how the number of a global variable can be used in a string:

```
  (assign, reg1, "$g_your_global_variable"),
  (str_store_string, s1, "@The value of the global variable is {reg1}."),
  (str_store_string, s0, s1),
```

[29] Bundeling notes of jik, Updated Module System Tutorial, page 20; WookieWarlord, Modding Q&A and Somebody, Modding Q&A, Modding Q&A.
[30] kalarhan, Modding Q&A.
[31] Somebody, Modding Q&A.
[32] Swyter, Modding Q&A.

As you might already be aware of, a @ begins any quick string which you haven't declared in module_strings.py.

{reg#?str1:str2} tests if reg# is true, and if so, displays str1. Otherwise it displays str2. The value of reg# is here a simply boolean with 1 being true and 0 being false.

5.2.3.2 Gender/Skin-based Alternations

Sort out informations given above, currently a mixmax subchapter.

One notable feature of strings is the ability to put a register value or even another string inside of it. You can do this by adding, for example, reg0 in the string. This will display the current value of reg(0). reg10 would display the current value of reg(10), and so on.

For additional strings, you must use the string register instead of a normal register; this is because strings are stored separately from registers. For example, s2 would display the contents of string register 2 – not the contents of reg(2). You can freely use string register 2 and reg(2) at the same time for different things, they will not overlap or interfere with each other.

In dialogue, it is also possible to display parts (or all) of a string's contents differently depending on the gender of the person being addressed. For example, inserting sir/madam into a string will cause the word "sir" to be displayed when the addressed person is male, and "madam" if the person is female.

All of these options (except gender-based alternation) will work perfectly in any kind of string field. Gender-based alternations only work in dialogue.

For the purposes of the operation `display_message`, it is possible to display a string in various colours, by appending a hexadecimal colour code to the operation. For example, (display_message,<string_id>,[hex_colour_code]),. For the colour codes see Subchapter 7.3.4.

Gender at Dialogues

Gender issue dialogues (read carefully and think about improvements at MS)[33]

new race gender problem [34]

gender true false dialog[35]

gender of new races[36]

Third option at gender decission not possible and chaining[37]

gender dialog[38]

[33] dunde, Modding Q&A.

[34] Somebody, Modding Q&A.

[35] Somebody, Modding Q&A.

[36] Lumos, Modding Q&A.

[37] Lumos, Modding Q&A and Modding Q&A, and Somebody, Modding Q&A.

[38] gsanders (credits), Modding Q&A.

gender new races[39]

gender of skins[40]

male female strings[41]

5.2.3.3 Difference between Strings and Quick Strings

Quick strings are `"@strings"`, while normal strings are `"str_strings"`. At compiling quick strings are put into quick_strings.txt but are not getting referenced like strings are. Why would you ever use quick strings over strings? Why doesn't the Module System just put it into strings instead? It is obvious that for momentary message displays quick strings are much more useful, as they can be defined on-the-fly. At the strings in module_strings.py the order (and indexes) are important since some scripts match some other id (for example a scene id) with the string id offset the same distance from the start. Lists can only be made with strings. Both, strings and quick strings, are getting translated though. Internally, there is no difference between strings and quick strings.[42]

str_ can be reffered to, @ can't. It's literally the only difference. If you are planning "fire and forget"text, use @. If it will participate in other fetures later, save as str.[43]

5.2.3.4 Other notes

No string comparision[44]

max lenght for strings[45]

Hardwired strings[46]

make variable to string[47]

store string in global var[48]

Convert value to a string[49]

[39] kalarhan, Modding Q&A.

[40] kalarhan, Modding Q&A.

[41] Multiple participants, Male races.

[42] Summary of Discussion by Lumos, cmpxchg8b and Vornne, Modding Q&A.

[43] Dalion, Mount & Blade Modding Discord.

[44] The_dragon, Modding Q&A.

[45] cmpxchg8b, Modding Q&A.

[46] Lav, Modding Q&A.

[47] Arch3r, Modding Q&A.

[48] kalarhan, Modding Q&A.

[49] kalarhan, Modding Q&A.

5.2.4 Module Scripts

5.2.4.1 Game Engine Scripts

hard-coded scripts[50]

Not going into details, short description of content and linking to Tutorial and Code Example Sections game_get_troop_wage : for weekly wage of troop game_get_join_cost : for cost when joining game_get_upgrade_cost : for cost of upgrading

Some special behaviour of game_get_skill_modifier_for_troop[51]

5.2.4.2 Can-fail (cf_x) Scripts

You have either noticed some scripts in module_scripts.py with the prefix `cf_` or gottem some warning at compiling your module with the following content:

```
WARNING: Script can fail at operation #x. Use cf_ at the beginning of its name:
    some_script_name
```

The prefix cf_ indicates a script that can fail. MaBL/MBScript requires that every command (line) is true or it jumps to the next block (scope). cf_-scripts are a handy way to use this:[52]

```
1  top level scope
2
3          | second level scope
4          |
5  (try_begin)
6          (call_script, "cf_something"),  # goes to line 2 (true) or 4 (false)
7          (assign, ":true", 1),
8          (display_message, "@ololol"),
9  (else_try),
10         (assign, ":true", 0), # if cf_ fails (is false)
11 (try_end),
```

You can read the warning as a recommendation, since you have a conditional operation that would cause the script to fail within the top level scope of the script rather than inside a try block. Therefore, if that operation causes the script to fail, the scope where the call_script operation calls this script

[50] MadocComadrin, Modding Q&A.

[51] cmpxchg8b, Modding Q&A and [WB] Warband Script Enhancer v3.2.0.

[52] kalarhan, Modding Q&A.

would also fail.[53] cf_-scripts are usually used as complex conditionals, you can use them to write your own conditional functions.[54]

To give a short example for a complex conditional: cf_-script can be used as a part of a this_or_next operation:

```
Example 1:
(this_or_next|eq, 1, 0),
(call_script, "script_cf_eq_1_1"),

Example 2:
(this_or_next|call_script, "script_cf_eq_1_1"),
(eq, reg1, 0),
```

Example 1 will work as a cf_-script can be used as the last case. Example 2 on the contrary does not work as intended, it just fails or continues as with a normal call_script operation.[55]

Coming back to the warning message mentioned above. Two ways have been now indicated on how to deal with it. The first option is to add the prefix cf_ infront of the script name, not because you simply want the warning message to disappear but because you understood the effect of it and how this script might affect your work. The second option is nearly as simple: If you don't want your script to behave like a can-fail one, wrap the entire script code between `try_begin` and `try_end`.[56]

5.2.4.3 Other Tidbits

Scripts called by the game engine are getting read top to bottom, so only the first script will be called in case there are numerous scripts with the same name. The script numeric ID is getting created at the compilation: the compiler loops through all scripts, starting from the first script (with the numerical ID 0), and compares the script identifier for which we are looking (`"script_abcd"`) with each script's string id. Once one occurrence is found, the numeric id of that script is returned, so the search stops and will never return the second or third occurrence of a script.[57]

Recheck this one, perhaps useful info[58]

[53] This means that scripts (or triggers) calling this cf_-script at the top level scope can also fail.

[54] Caba'drin, Modding Q&A, and Somebody, Modding Q&A.

[55] Vornne, Modding Q&A and Modding Q&A.

[56] The_dragon, Modding Q&A.

[57] The_dragon, Modding Q&A.

[58] RecursiveHarmony (credit), Script file unknown float variable.

5.2.5 Module Simple Triggers

```
Each simple trigger contains the following fields:
1) Check interval: How frequently this trigger will be checked
2) Operation block: This must be a valid operation block.
```

There are two types of triggers at the M&B game engine: Regular triggers and simple triggers. We will begin with the simple triggers because they are generally smaller and easier to understand.[59]

Before that though, what is a trigger? A trigger is a block of code that executes multiple times as the game progresses. It's called a trigger, because it fires - that is how it's called when the time comes for the trigger to do its job.

Simple triggers are the alternative to old style triggers. They do not preserve their state and are thus simpler to maintain. Simple triggers are generally found in module_simple_triggers.py and despite being simpler, they are not to be underestimated. Let's have a look at a simple trigger:

```
(24, [
  (troop_add_gold, "trp_player", 100),
  ]),
```

This is rather simple, isn't it? Every 24 hours, so every day at the world map, this trigger will give 100 gold to the player. Yes, it makes no sense, but it works. And that is basically how every simple trigger operates. Here is the structure: (check_interval, [operations_block]),

This is shared by each and every simple trigger, including the gold-giving one which we have got above. Now let's look at the components: The **check interval** is how frequently this trigger will be checked, measured in in-game hours. Using 24 will make the trigger fire once every day, using 1 will make it fire 24 times per day. Using 0 as a check interval will make the trigger fire once every frame, which can be many times per second, dependending on the user's frame rate. Such triggers are potential performance hogs, so be cautious when using them.

The **operations block** is what happens when the trigger fires, and can contain whatever codes you want to place in it. Yes, checks are allowed. For an example, we can enhance our aforementioned code by doing this:

```
(24, [
  # (try_begin), # We don't need try blocks right now
  (lt, "$player_honor", 0),
  (troop_add_gold, "trp_player", 100),
  # (try_end),
  ]),
```

[59] The upfollowing paragraphs are taken from Lumos, Modding Q&A.

Now it will give a hundred gold per day to the player, if he/she is dishonourable. It makes even less sense now, but whatever. As you can see, a trigger can fail with no problems whatsoever, however be careful when not wrapping statements in try_ blocks. Of course, it's always safer to use the try blocks, just in case.

```
(24, [
  (try_begin), # Better safe than sorry
    (lt, "$player_honor", 0), # Make sure your indentation is always correct!
    (troop_add_gold, "trp_player", 100),
  (try_end),
  ]),
```

Right, so we've got the basics covered now. There are more things you need to know though. Simple triggers are also used in other places, such as on items and scene props, but with special conditions in form of hard-coded conditions passed by the engine at the right times, like for example `ti_on_weapon_attack` or `ti_on_init_scene_prop`. A list of all such *special* check intervals can be found in header_triggers.py.

This sums up pretty much everything related to simple triggers, so you can move on to the next chapter about triggers.

5.2.5.1 Save-game Compatibility

save-game compatibility.[60]

5.2.5.2 Deprecated Hardcoded Triggers

ti_on_party_encounter and ti_simulate_battle fully deprecated or still useable (given it makes sense to use them)? Two game scripts get active at the trigger moment anyway.

switch to world map trigger perhaps simple trigger specific[61]

[60] cmpxchg8b, Modding Q&A.

[61] kalarhan, Modding Q&A.

5.2.6 Module Triggers

```
Each trigger contains the following fields:
1) Check interval: How frequently this trigger will be checked.
2) Delay interval: Time to wait before applying the consequences of the trigger.
   After its conditions have been evaluated as true.
3) Re–arm interval: How much time must pass after applying the consequences of the trigger for the trigger to become active again.
   You can put the constant ti_once here to make sure that the trigger never becomes active again after it fires once.
4) Conditions block (list): This must be a valid operation block. See header_operations.py for reference.
   Every time the trigger is checked, the conditions block will be executed.
   If the conditions block returns true, the consequences block will be executed.
   If the conditions block is empty, it is assumed that it always evaluates to true.
5) Consequences block (list): This must be a valid operation block. See header_operations.py for reference.
```

The regular triggers, called only triggers, are a bit more complex and versatile than their simpler counterparts, but are largely the same. This kind of triggers is however more important because they are also getting used in the mission templates, about which we will talk in a later chapter. Triggers within module_triggers.py fire however only at the world map. Let's examine the structure of a trigger: (check_interval, delay_interval, re−arm_inverval, [conditions_block], [operations_block]),[62]

As you can see, there are more fields present here than there are at the simple triggers. Now let's examine them one by one.

- The **check interval** is absolutely the same as before - how often the trigger fires;
- The **delay interval** is how much the consequences should be delayed once the conditions are true;
- The **re-arm interval** is how long the trigger should remain inactive once the consequences are applied;
- The **conditions block** contains the code that needs to be checked in order to allow for the consequences to fire;
- The **consequences block** is where the majority of your ocde is usually located, it will be executed once the conditions are evaluated to true and the delay interval wears off.

[62] The upfollowing paragraphs are taken from Lumos, Modding Q&A.

Now, how would our senseless simple trigger from the chapter before look like as a regular trigger?

```
(24, 0, 0, [], [
  (try_begin),
    (lt, "$player_honor", 0),
    (troop_add_gold, "trp_player", 100),
  (try_end),
  ]),
```

This is perfectly valid, but it's frankly kind of stupid if we are going to use a regular trigger. Let's buff up the rewards and rework the code a bit in order to make use of the systems that this trigger allows us:

```
(24, 6, 48, [
  (lt, "$player_honor", 0),
  ], [
  (troop_add_gold, "trp_player", 400),
  ]),
```

Now, every 24 hours this trigger will evaluate the player's honour. If it's below zero, six hours will pass, keeping the player on edge (okay, not really), and then he or she will be awarded four hundred gold. However, the trigger will not fire for the next 48 hours, which kind of compensates for the greater reward. Yes, the trigger is still dumb, but it serves as a nice example.

Regular triggers also have (in other files than module_triggers.py) their special conditions, most of which related to missions like for example `ti_before_mission_start` or `ti_on_agent_spawn`, once again listed in header_triggers.py.

5.2.6.1 Remarkable Notes

1. The check, delay and re-arms interval of triggers within module_triggers.py are in hours and of triggers within module_mission_templates.py they are in seconds. Setting a zero as check interval lets the trigger however fire at every frame, it is not affected by the party movement or difference in time flow. It even runs when your party stops and the world map pauses.[63]

2. Avoid too many specific triggers[64]

3. Local variables are not transferred between conditions and consequences. In other words, you will need to store a variable again to modify it in the consequences if you have already stored it to check it.

[63] kalarhan and RecursiveHarmony, Modding Q&A.

[64] kalarhan, Modding Q&A.

4. When dealing with triggers with special conditions, you are likely to need to check their parameters using the operations `store_trigger_param_*`. Consult header_triggers.py for a list of parameters for each trigger.

5.2.6.2 Randomness of Triggers

(simple) triggers a bit random[65]

5.2.6.3 Randomly Delayed Trigger[66]

Now you know that the check intervals at triggers already fire randomly. What now if you want a randomly delayed trigger instead? Say, instead of a trigger that fires once after 24 hours, like this one

```
(2, 24, ti_once, [], [<your code>]),
```

you want one that fires once but randomly delayed between 24 and 48 hours after it got triggered? In that case you should use two triggers and global variables to do it, like you can see in the following example

```
(2,0,ti_once, [(store_random_in_range, "$g_delayed", 24, 49), (assign, "$g_run", 0),], []),

(1,0,ti_once, [(val_add, "$g_run",1), (gt,"$g_run","$g_delayed"),], [<your code>]),
```

[65] Leonion, Modding Q&A.

[66] dunde, Modding Q&A.

5.2.7 Module Factions

```
Each faction record contains the following fields:
1) Faction id: used for referencing factions in other files.
   The prefix fac_ is automatically added before each faction id.
2) Faction name.
3) Faction flags. See header_factions.py for a list of available flags
4) Faction coherence. Relation between members of this faction.
5) Relations. This is a list of relation records.
   Each relation record is a tuple that contains the following fields:
  5.1) Faction. Which other faction this relation is referring to
  5.2) Value: Relation value between the two factions.
       Values range between -1 and 1.
6) Ranks
7) Faction color (default is gray)
```

module_factions.py contains all the factions used by the module system and by M&B's artificial intelligence. Though a small file, it governs several important settings which we will cover here.

The file immediately begins with a Python list: factions = [. As in most of the module files, the first few tuples are hardwired into the game and should not be edited. If you scroll down just a bit, you will find the "deserters" tuple as an example of a faction:

```
("deserters","Deserters", 0, 0.5,[("manhunters",-0.6),("merchants",-0.5),("player_faction",-0.1)], [], 0x888888),
```

This is a short, simple tuple. It governs the "deserters" faction, whose major enemies are the "manhunters", "merchants", and "player_faction" factions. If a faction's relation with "deserters" is not defined anywhere in module_constants, it will automatically be set to 0, hence the two factions will be absolutely neutral to each other.

"deserters" tuple examination:

```
1) Faction id = "deserters"
2) Faction name = "deserters"
3) Faction flags = 0
4) Faction coherence = 0.5
5) Inter-faction relations:
    5.1) Faction = "manhunters", "merchants", "player_faction"
    5.2) Relation = -0.6, -0.5, -0.1
6) no Rank is listed
7) the color of this faction is set the to the hex value of 0x888888 (Just like with strings, only the last 6 digits are the RGB values)
```

This faction has no flags, a neutral faction coherence, and three enemies defined in its tuple. However, if we look elsewhere, we can find that (for example) all "kingdom_#"s have a relation of -0.02 with "deserters", as defined in any of the kingdom tuples (such as "kingdom_1", the "Kingdom of Swadia"). This is because a relation works two ways – it only needs to be set once, and the value will count for both factions. In general, the value 0 means that the factions are at peace and a negative value means that those factions are at war. Positive numbers do not really matter with Native diplomacy.[67]

It should also be possible to declare a constant like default_kingdom_relations before the faction list is beginning in which some often used relationships are getting defined and which are then replacing some of the tuples.

Regarding field 6, "Ranks": It is a vestigial feature from the M&B versions 0.7xx and earlier - you could earn faction reputation from quests and earn stipends from being a retainer. Instead of being a vassal holding land and such, you used to get paid by advancing in ranks.[68] It is now deprecated and not getting used anymore.

5.2.7.1 Difference between Faction and Culture

In Native you will discover at some point that there exist factions and cultures. The culture setup basically allows a kind of meta-faction to exist. For example, if the Kingdom of Swadia is wiped out, their culture (Swadian) still exists in their original villages and towns. The culture is not a kingdom - just a placeholder of sorts for the basic tier 1-5 troops, town walkers, etc.[69]. A culture is thus basically just a reference to a specific troop tree since it is mostly used at the recruitment to define the type of recruits which depend on the culture of the village. However, if you search for `culture_1` at the files you will also find the script "get_culture_with_faction_for_music". This script assigns a music track to each faction, in our example to Kingdom 1, assigning the flag `mtf_culture_1` which has been defined in header_musics.py and been set up in module_music.py. At the world map and at the scenes the player hears at the respective locations the music tracks for this culture. Those flags represent a soft limit for the number of different cultures since you cannot define more than six. However, as long as you ignore the part with the music tracks[70] you can setup as many cultures as you want.

5.2.7.2 Changing Faction Relations[71]

The most direct method would be to use the operation `set_relation` but that doesn't taken into account the various political consequences. It is therefore recommended to look in module_scripts.py

[67] Caba'drin, Modding Q&A.
[68] Somebody, Modding Q&A and Modding Q&A.
[69] Somebody, Modding Q&A.
[70] For which you can find workarounds like they can be found for e.g. at the Mod "The Last Days of the Third Age".
[71] Summary of posts by Somebody and Ikaguia, Modding Q&A.

for its usage. Take note that when you change Faction A relation with Faction B to X, you are automaticly changing Faction B relation with Faction A to X too. The relation should be between 1 (friendly) and -1 (hatefull) with 0 being neutral. Anything less than 0 will cause atacks with a value of -1 causing the most atacks.

5.2.7.3 Faction Flags

ff_always_hide_label: This flag is getting used by the "Innocents" and "Merchants" factions. It hides the faction name when hovering with the mouse over any party icon belonging to that faction in the 'popup tooltip'.

5.2.7.4 Differences between Player's Faction and Player Supporters' Faction

The similar sounding constants `fac_player_faction` and `fac_player_supporters_faction`, and the variable `$players_kingdom` might be irritating at first sight. Latter one is a variable which will hold different values depending on the what the player is doing in the game. If a player is a vassal of a kingdom, `"$players_kingdom"` will hold the faction ID (`"fac_kingdom_X"`) value of that faction. The `"fac_player_supporters_faction"` is the actual faction that gets assigned to `"$players_kingdom"` and to the player's vassals if the player is a rebel or creates their own faction. The `"fac_player_faction"` tracks player-specific relations with other factions, etc.

5.2.7.5 Maximum Amount of Factions

There is a hard-coded limit of 128 factions, 0 to 127 (signed small int). You can't bypass the engine with Python hack.[72]

[72] kalarhan, Modding Q&A.

5.3 Agent-related Module System Files

5.3.1 Module Skills

```
Each skill contains the following fields:
1) Skill id (string): used for referencing skills in other files. The prefix skl_
    is automatically added before each skill-id .
2) Skill name (string).
3) Skill flags (int). See header_skills.py for a list of available flags
4) Maximum level of the skill (int).
5) Skill description (string): used in character window for explaining the skills.
```

module_skills.py contains all the skills used by the module system. A skill grants an ability or buff to a character or a party. The level of the skill determines its potency. In Native each of a character's skills can be increased from level 0 to level 10 and there are 24 skills.

Each skill has a base attribute. A character's level of a skill cannot surpass one third of their level in its base attribute (except through the use of Books). For example, Tactics has Intelligence as its base attribute; if Intelligence is level 9, Tactics cannot surpass level 3. For players, initial skill levels are dictated by their character's background and may exceed this limitation.

The type of a skill determines how it affects the party. There are three types of skills:

Party skill A party skill grants an ability to the party as a whole. The level of a party skill is chosen from the party member who has the highest level in that skill, regardless whether it is the player or a companion. If the player has a party skill trained to level 2 or higher, a bonus may be awarded (see below).

Leader skill Like a party skill, a leader skill grants an ability to a party. However, only the party-leader's level in the skill is used.

Personal skill A personal skill grants an ability to an individual party-leader or companion. In general this ability only benefits the individual. However, in the case of 'Trainer', the whole party benefits from bonus experience. Therefore multiple companions with 'Trainer' can be used to give a greater advantage to the party.

Party Skill Bonuses If the player character has a sufficient level in a party skill, a bonus will be awarded. If the party's highest skill is not from the player character, the member who has the highest level in that skill will determine the party skill base value, while the player character's level will grant level bonuses according to the chart below. This means that party skills can obtain a maximum level of 14 when the player character is trained up to level 10. Once this level of skill is achieved, other members of the party with the same skill only serve a backup role in case the party leader falls below the health threshhold for their skills to be active.

Skill Level	Bonus
0-1	(+0)
2-4	(+1)
5-7	(+2)
8-9	(+3)
10	(+4)

5.3.1.1 Native Skills and Effects

```
0)  Trade (Party skill): Every level of this skill reduces your trade penalty by 5 %.
1)  Leadership (Leader skill): Every point increases maximum number of troops you can command by 5, increases your party morale and reduces troop wages by 5 %
2)  Prisoner Management (Leader skill): Every level of this skill increases maximum number of prisoners by 5.
7)  Persuasion (Personal skill): This skill helps you make other people accept your point of view. It also lowers the minimum level of relationship needed to get NPCs to do what you want.
8)  Engineer (Party skill): This skill allows you to construct siege equipment and fief improvements more efficiently.
9)  First Aid (Party skill): Heroes regain 5 % per skill level of hit-points lost during mission.
10) Surgery (Party skill): Each point to this skill gives a 4 % chance that a mortally struck party member will be wounded rather than killed.
11) Wound Treatment (Party skill): Party healing speed is increased by 20 % per level of this skill.
12) Inventory Management (Leader skill): Increases inventory capacity by +6 per skill level.
13) Spotting (Party skill): Party seeing range is increased by 10 % per skill level.
14) Path-finding (Party skill): Party map speed is increased by 3 % per skill level.
15) Tactics (Party skill): Every two levels of this skill increases starting battle advantage by 1.
16) Tracking (Party skill): Tracks become more informative.
17) Trainer (Personal skill): Every day, each hero with this skill adds some experience to every other member of the party whose level is lower than his/hers. Experience gained goes as: {0,4,10,16,23,30,38,46,55,65,80}.
```

```
22) Looting (Party skill): This skill increases the amount of loot obtained by 10 % per skill level.
23) Horse Archery (Personal skill): Reduces damage and accuracy penalties for archery and throwing from horseback.
24) Riding (Personal skill): Enables you to ride horses of higher difficulty levels and increases your riding speed and manuever.
25) Athletics (Personal skill): Improves your running speed.
26) Shield (Personal skill): Reduces damage to shields (by 8 % per skill level) and improves shield speed and coverage.
27) Weapon Master (Personal skill): Makes it easier to learn weapon proficiencies and increases the proficiency limits. Limits go as: 60, 100, 140, 180, 220, 260, 300, 340, 380, 420.
33) Power Draw (Personal skill): Lets character use more powerful bows. Each point to this skill (up to four plus power-draw requirement of the bow) increases bow damage by 14 %.
34) Power Throw (Personal skill): Each point to this skill increases throwing damage by 10 %.
35) Power Strike (Personal skill): Each point to this skill increases melee damage by 8 %.
36) Ironflesh (Personal skill): Each point to this skill increases hit points by +2.
```

5.3.1.2 Hard-coded Skills

There exists a list of hard-coded skills which are (indexed, beginning with 0): Trade (0), Leadership (1), Prisoner Management (2), Engineer (8), First Aid (9), Surgery (10), Wound Treatment (11), Inventory Management (12), Spotting (13), Pathfinding (14), Tactics (15), Tracking (16), Trainer (17), Horse Archery (23), Riding (24), Athletics (25), Shield (26), Weapon Master (27), Power Draw (33), Power Throw (34), Power Strike (35), Ironflesh (36).

The effects of these skills can only be removed if the skill is disabled with sf_inactive flag.[73] If you want to add a new skill, use the reserved skills or use soft-coded skills.

5.3.1.3 Skill Flags

sf_base_att_str: This flag sets Strenght as base attribute of this skill.

sf_base_att_agi: This flag sets Agility as base attribute of this skill.

sf_base_att_int: This flag sets Intelligence as base attribute of this skill.

[73] Khamukkamu however remarks that hiding any of the default skills will cause the game to crash. It is better to just rename the skill to −−. Mount & Blade Modding Discord.

sf_base_att_cha: This flag sets Charisma as base attribute of this skill.

sf_effects_party: With this flag the skill grants an ability to the party as a whole.

sf_inactive: Disables a skill.

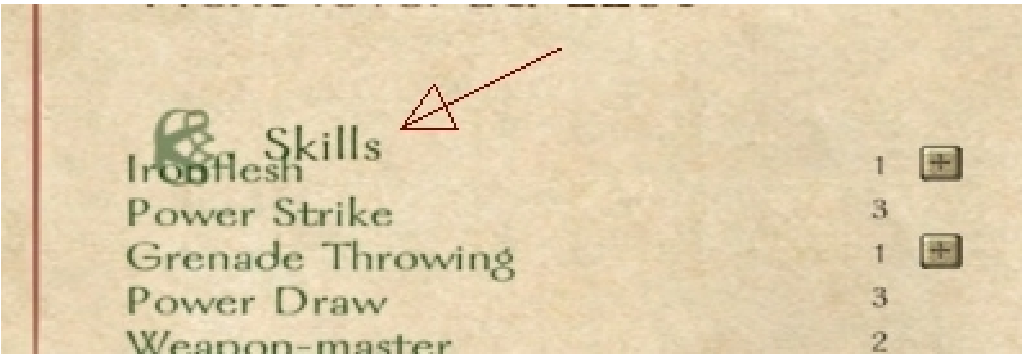
5.3.1.4 Maximum Skill Level

Skills are stored in 4 bits - that is to say, limited to 15 (it's higher for attributes). If you look properly at header_skills.py, you will see that every skill increment is a multiple of the first (which is a power of 16) - this is so that all possible skill combinations can be fit into a single field.[74] However, in Native the maximum skill level is at 10 since it is missing the definitions for the upper ones. You need to define them by yourself or make use of a more expanded version of the Module System.

5.3.1.5 Adding, Combining, Deleting and/or Editing Skills

You can add a new skill by using one of the 18 reserved skills of which you can edit the skill ID and name as you wish. Keep in mind that it is not possible to add more skills,[75] so you are restricted to 42 skills in total.

If you add a new skill you might get irritated by your character window in-game since the skill list is expanding and thus moving up, ending in a result like in the image below. The easiest way to fix this is to remove some other skill but that is in most cases not something you would like to do. Although the character window is a hard-coded menu you have some possibilities to alter the panel itself. The background image is stored as mesh `character_window` in one of the `CommonRes/core_*.brf` files. You can create your own resource file declaring the same mesh with a different texture and it should override the default one. Or you can just copy the texture `character_window.dds` to your module's Textures folder and edit that one to your heart's content.



Skills
Ironflesh 1
Power Strike 3
Grenade Throwing 1
Power Draw 3
Weapon-master 2

[74] Somebody, Modding Q&A.
[75] cmpxchg8b, [WB] Warband Script Enhancer v3.2.0.

Once you have enough space on the background, you will need to edit the text and/or elements positioning. The coordinates for them are stored in the file `game_variables.txt` - they are annotated and should be easy to find. Just keep in mind that base x/y coordinates and container sizes are usually absolute (with screen having 1.0 * 0.75 dimension, no matter what aspect ratio it actually has), while element/text size values are usually relative (so width 2.0 doesn't mean "twice wider than screen"but "twice wider than element's mesh as measured by ruler in OpenBrf").[76]

You might want to combine, edit or delete some of the Native skills and ask yourself if that is possible. It depends, most skills have hard-coded effects while some other effects are defined in the Module System. Latter one can easily be edited. For the hard-coded ones you would need to reverse engineer the effects if possible and then apply your own effects. For example, the skills `Spotting` and `Pathfinding` can be simulated via the game scripts `game_check_party_sees_party` and `game_get_party_speed_multiplier` while `Tracking` is a skill which can not be simulated.[77] When checking for a skill effect you have three options:

1. hard-coded stuff (will require extra work to override)
2. hard-coded stuff but with open settings on module.ini (so just tweak the flag/variable there)
3. Module System effect (open code)

Start by finding the reference (key). It is a skill, so go to `module_skills.py`. For skills you add `skl_`, so the code reference is `skl_leadership` for the key `Leadership`, etc. Now do a full text search for this on your Module System and check where it is getting used. To stay with the example, you would eventually find the script `game_get_party_companion_limit` that implements the logic around party size limit. Note also how the script name normally tells you what it is used for.[78]

[76] Lav, Modding Q&A.

[77] _Sebastian_, Ikaguia and Somebody, Modding Q&A.

[78] kalarhan, Modding Q&A.

5.3.2 Module Skins

```
Each skin record contains the following fields:
1) Skin id: used for referencing skins.
2) Skin flag: skin will use this morph key/frame of a mesh if multiple are given.
3) Body mesh.
4) Calf mesh (left one).
5) Hand mesh (left one).
6) Head mesh.
7) Face keys (list)
8) List of hair meshes.
9) List of beard meshes.
10) List of hair textures.
11) List of beard textures.
12) List of face textures.
13) List of voices.
14) Skeleton name
15) Scale (doesn't fully work yet)
16) Blood particles 1 (do not add this if you wish to use the default particles)
17) Blood particles 2 (do not add this if you wish to use the default particles)
18) Face key constraints (do not add this if you do not wish to use it)
```

module_skins.py is where all the skins for the troops and heroes are defined. They are getting used for genders as well as for different races, so whenever you wish to make a new skin for troops, this is the file you will be modding. You can have a maximum of 16 different skins which is given by header_troops.py and cannot be increased:[79]

```
troop_type_mask = 0x0000000f
tf_hero = 0x00000010
```

If each race has a male and female version, then there will be 8 races. Take note that in the multiplayer you can have only two races because the netcode only uses 1 bit for skin[80] which are usually used for the male and female skin.

5.3.2.1 Adding a New Skin

Note that you don't need a new race unless you want to change any of the following: body mesh, face shape, war cries, death noises, blood effects. The rest can be done really easily with extra face textures and careful face code usage. If you are adamant about having different face shapes, be aware

[79] dunde, Modding Q&A.
[80] cmpxchg8b, Modding Q&A.

that you would have to make the meshes and vertex animations yourself, fit the beards to the new mesh, and make a bunch of new textures with different mapping.[81]

In the Native Module System you can either reactivate the skin “undead“, by removing the # in front of it, or you add following lines beneath it (but before last]):[82]

```
(
  "<race>", 0,
  "<race>_body", "race_calf_l", "race_handL",
  "<race>_head", race_face_keys,
   [],
   [],
   [],
   [],
  [("<RACE>face_a",0xffffffff,[]),
   ("<RACE>face_b",0xffcaffc0,[]),
    ], #face textures
   [], #voice sounds
  "skel_human", 1.0,
),
```

Change all `<race>` to the name of your new race and models. The models which are refered here, are stored in body_meshes.brf. skel_* referes to the skeleton for the race. If you want to add Dwarves, Goblins, Hobbits, Giants or something like this, you will probably want to change this and need a custom skeleton. Afterwards add in header_troops.py the new skin flag. You will see there the following entries:

```
#Troop flags
tf_male             = 0
tf_female           = 1
##tf_undead            = 2
```

When adding a new race, you should add them here after the commented-out undeads (or replacing that line). You need to remember that the number which follows the skin must be unique (the undeads are commented out, so you can use 2). So it should look like:

```
tf_dwarf            = 2
tf_hobbit           = 3
```

Now you can use the new troop flag for all troop entries which should have the new skin. Next, if you want race to be selectable for player, go to module_game_menus.py and search for ("start_male". under this:

[81] jacobhinds, Modding Q&A.

[82] Following here onward a post of lllew, Modding Q&A.

```
        ("start_female",[],"Female",
         [
           (troop_set_type, "trp_player", 1),
           (assign, "$character_gender", tf_female),
           (jump_to_menu, "mnu_start_character_1"),
         ]
         ),
```

but before ("go_back",[],"Go back", you have to add something like this:

```
        ("start_<race>",[],"<Race>",
         [
           (troop_set_type, "trp_player", 2),
           (assign, "$character_gender", tf_<race>),
           (jump_to_menu, "mnu_start_character_1"),
         ]
         ),
```

Note that the `2` after `"trp_player"` refers to the number which you used in header_troops.py.

5.3.2.2 Skin Flag and Morph Key

Tuple field 2, Skin Flag, determines which morph key is getting used for that skin. Basically you can use this bit to tell the game if the skin (in most cases a race) requires some body transformation, in the same way as the female armour has narrower shoulders and breasts. If you open up openbrf and look at an armour which features a feminised form you will notice that the female form will be in frame number two with time 10. To tell the game to use this vertex animation frame and time for a particular race you use `skf_use_morph_key_10` as the time of the frame is 10. Following the same logic you can create your own new frames for particular races and use `skf_use_morph_key_20` up to `skf_use_morph_key_70`.[83] The available flags at header_skins.py determine that you cannot have more than eight morph keys for the frames.

While you should use a different vertex animation frame for each race that use different skeleton (you can use OpenBRF to reskeletonize the armors and add the frames, see the chapters *missing* and 12.2.4.)[84] the game engine only makes use of the morph key if such a frame is given. If you assign an armour[85] to a troop of a race with a different skeleton which has not such vertex animation frame given, the game will use the default one.

5.3.2.3 Face Keys Code

Each face code consists of hexadecimal numbers in a specific order that define all the traits of a person (hair color, age, beard type etc. etc.), with the traits having a range (let's say 00 would be almost closed eyelids, ff widely opened eyelids, and then everything in between).[86]

face codes explained[87]

documentation face key string[88]

You can get a face that ages by giving the material of your face an old version texture as diffuseB and old version normal map as specular.[89]

You might get an issue where faces are white from a distance but turn normal at closer distance. This is because you need a .LOD material (note the capitals) as well as meshes. Look at how everything else is set up in materials_face_gen.[90]

If you are using a face key on a register note that it doesn't use the 0x (also remember to set

[83] La Grandmaster, Modding Q&A.

[84] dunde, Modding Q&A.

[85] Morph keys work only on body armors (boots and helmet only with WSE); cmpxchg8b, Modding Q&A.

[86] Lord Szentgyorgyi, Modding Q&A.

[87] Yoshiboy, random face code generator.

[88] Dusk Voyager, Modding Q&A, and k61824, Modding Q&A.

[89] dunde, Modding Q&A.

[90] Somebody, Modding Q&A and Modding Q&A.

both keys or use the default operation).[91]

```
(str_store_string, s1, "
    @000000000a00000136db6db6db6db6db00000000001db6db0000000000000000"),
```

5.3.2.4 Skin Colour

Posts to integrate: Yoshiboy, Modding Q&A;

Vertex painting[92]

skin colours troops[93]

skin colour[94]

vertex colour body[95]

forearm vertex colour[96]

skin coloured meshes[97]

face colour codes[98]

Other materials for hair[99]

some module skins explanation[100]

face key registers[101]

Some skin stuff[102]

5.3.2.5 Scaling Problems

Tuple field 15, Scale, is an interesting but also sadly not fully functional field. The scale factor multiplies everything, so the skeleton, the hitboxes and all meshes getting used by troops with the respective skin flag. It seems to be thus an easy useable way to create Dwarves or Giants. It comes however with some drawbacks: If the troop gets hit by missiles, the missiles sticking in the troop scales with the same factor.[103] The walk cycles for shorter races are faster while the one for bigger races is slowier which is in most cases wanted. Also it might come to an scale up bug, a ragdoll issue, when the size is set to an factor bigger than 1. That one might however be fixable by delaying

[91] kalarhan, Modding Q&A.
[92] GetAssista, Modding Q&A.
[93] Somebody, Modding Q&A.
[94] Somebody, Modding Q&A.
[95] Somebody, Modding Q&A.
[96] Somebody, Modding Q&A.
[97] jacobhinds, Mount & Blade Modding Discord.
[98] Leonion (credit), Modding Q&A.
[99] xPearse, Modding Q&A.
[100] snouz (credit), Modding Q&A.
[101] cmpxchg8b, [WB] Warband Script Enhancer v3.2.0.
[102] jcsm130988, Body-Head Color Tone Mismatch.
[103] DtheHun, Modding Q&A.

the time the death agent turning into a ragdoll on any death animations[104] or fixing the associated hitboxes.[105]

For a more detailed explanation it is recommended to read the report of scale experiments performed by Khamukkamu which contained a variety of tests, explanations and summaries.[106] It is also recommended to look up how the scale factor is getting used in *Viking Conquest*: the developers used their largest character (Tall) to be of the standard (scale = 1) size. From there, they reduced the scale of each skin (something like normal height=0.9, short=0.85, child=0.75). Beside the faster walk cycles for the shorter/normal characters they found thus a good way around the other above-mentioned issues.

5.3.2.6 Skins for Horses

You cannot add a new horse skeleton[107]. However, you can vary its scale. Use for this in **module_items.py** the entry `horse_scale(110)` for a bigger horse and `horse_scale(90)` for a smaller horse, you can experiment around with them. `horse_scale(30)` can for example be used to animate cats with the horse skeleton and animations.[108] Be aware that the set of animations is the same for every mount. It's however possible to find workarounds here as well as some mods removed for example the galloping by mounting the mount on one bone to get a proper futuristic motorbike.[109]

dynamic positioning of horse skeleton[110]

5.3.2.7 Bones List

list of bone id[111]

agent bones[112]

bones list[113]

hit detection[114]

Also reference to hit detection of engine subchapter.

order of bones cannot be changed.[115]

[104] dunde, Modding Q&A.
[105] Maelubiyet, Modding Q&A.
[106] Khamukkamu, Of Trolls, Orcs, and Elves: Experiments using SCALE from module_skins.py.
[107] GetAssista, Modding Q&A, and dunde, Modding Q&A.
[108] NPC99, Modding Q&A.
[109] dunde, Modding Q&A.
[110] Norwood Reaper, Modding Q&A.
[111] MitchPTI, Modding Q&A.
[112] _Sebastian_, Modding Q&A.
[113] Swyter, [WB] Warband Script Enhancer v3.2.0.
[114] Norwood Reaper, Modding Q&A.
[115] cmpxchg8b, Suggestions/requests thread – post yours here.

5.3.2.8 Blood Particles

open question about new races blood[116]

Blood on agents[117]

blood particle[118]

Remove blood of armour[119] or troop[120]

5.3.2.9 Skin/Gender/Race-based Voice Entries

Tuple field 13, the list of voices, is something which gets often missed when adding new skins.[121] Looking at the Native man skin entry, you can see the following sound entries for different occassions, like when the agent dies, gets hits, yells or makes a victory cheer:

```
[(voice_die,"snd_man_die"),(voice_hit,"snd_man_hit"),(voice_grunt,"snd_man_grunt"),(voice_grunt_long,"snd_man_grunt_long"),(voice_yell,"snd_man_yell"),(voice_stun,"snd_man_stun"),(voice_victory,"snd_man_victory")], #voice sounds
```

Compare it to the Native woman skin entry and you will discover that it has far less entries:[122]

```
[(voice_die,"snd_woman_die"),(voice_hit,"snd_woman_hit"),(voice_yell,"snd_woman_yell")], #voice sounds
```

A look into header_skins.py reveals which voice events can get used by modders:

```
voice_die        = 0    # Agent dies
voice_hit        = 1    # Agent gets hit
voice_grunt      = 2    # Agent swings a weapon
voice_grunt_long = 3    # Agent swings a heavy weapon(?)
voice_yell       = 4    # Agent
voice_warcry     = 5    # Agent
voice_victory    = 6    # Agent makes a victory cheer
voice_stun       = 7    # Agent gets stunned
```

[116] Khamukkamu, Modding Q&A.

[117] Jacobhinds, Putting blood on agents.

[118] K700, [WB] Warband Script Enhancer v4.8.0 for 1.174.

[119] GetAssista and Somebody, Modding Q&A.

[120] Specialist, Modding Q&A.

[121] The upfollowing notes are taken from nema, How to resolve problem with silent troops, updated by new infobits.

[122] This is also the reason why female victory cheering has often been reported as a problem, there is simply a `voice_victory` entry missing for it to work.

5.3.3 Module Troops

```
Each troop contains the following fields:
 1) Troop id (string): used for referencing troops in other files. The prefix trp_ is automatically added before each troop-id .
 2) Troop name (string).
 3) Plural troop name (string).
 4) Troop flags (int). See header_troops.py for a list of available flags
 5) Scene (int) (only applicable to heroes) For example: scn_reyvadin_castle|entry(1) puts troop in reyvadin castle's first entry point
 6) Reserved (int). Put constant "reserved" or 0.
 7) Faction (int)
 8) Inventory (list): Must be a list of items, at max. 64 items.
 9) Attributes (int): Example usage: str_6|agi_6|int_4|cha_5|level(5)
10) Weapon proficiencies (int): Example usage:
   wp_one_handed(55)|wp_two_handed(90)|wp_polearm(36)|wp_archery(80)|wp_crossbow(24)|wp_throwing(45)
    The function wp(x) will create random weapon proficiencies close to value x. To make an expert archer with other weapon proficiencies close to 60 you can use something like: wp_archery(160) | wp(60)
11) Skills (int): See header_skills.py for a list of skills. Example:
  knows_ironflesh_3|knows_power_strike_2|knows_athletics_2|knows_riding_2
12) Face code (int): You can obtain the face code by pressing ctrl+E in face generator screen
13) Face code (int)(2) (only applicable to regular troops, can be omitted for heroes): The game will create random faces between Face code 1 and Face code 2 for generated troops
14) Troop image (string): If this variable is set, the troop will use an image rather than its 3D visual during the conversations
```

module_troops.py is where all regular troops, heroes, chests and town NPCs are defined, complete with faces, ability scores and inventory. Whenever you wish to make a new character or troop type, this is the file you'll be modding.

The file begins with a small block of code that calculates weapon proficiencies and some other non-moddable code. Skip ahead to the list `troops = [`. Here we find tuples for the player and several other troops important to the game. Just below that are the various fighters we encounter in the arena fights. We'll study a few of these, as they are excellent examples of regular troops' level progression. Observe:

```
["novice_fighter","Novice Fighter","Novice Fighters",tf_guarantee_boots|tf_guarantee_armor,no_scene,reserved,fac_commoners,
[itm_hide_boots],
str_6|agi_6|level(5),wp(60),knows_common,mercenary_face_1, mercenary_face_2],
```

This is a bog-standard troop called "novice fighter". "novice fighter" is low-level, not very good at fighting, has low ability scores, and is otherwise unremarkable. Novice_fighter tuple examination:

```
1)  Troop id = "novice_fighter"
2)  Troop name = "novice_fighter"
3)  Plural troop name = "novice_fighters"
4)  Troop flags = tf_guarantee_boots|tf_guarantee_armor
5)  Scene = no_scene
6)  Reserved = reserved
7)  Faction = fac_commoners
8)  Inventory = [itm_sword,itm_hide_boots]
9)  Attributes = str_6|agi_6|level(5)
10) Weapon proficiencies = wp(60)
11) Skills = knows_common
12) Face code = swadian_face1
13) Face code 2 = swadian_face2
```

There are three things worth noting about this tuple.

Our "novice fighter" has tf_guarantee_armor, but no armour of his own. However, this does not make tf_guarantee_armor redundant; the troop will put on any armour he receives during the game.

To begin with (ie, at Level 1), "novice_fighter" has a STR of 6 and an AGI of 6. Upon start of the game, he is bumped up to level 5, with all the usual stat gains that implies.

He has the skill knows_common. knows_common is a collection of skills that was defined at the beginning of module_troops; scroll up in module_troops.py and observe this collection now.

```
knows_common = knows_riding_1|knows_trade_2|knows_inventory_management_2|
    knows_prisoner_management_1|knows_leadership_1
```

A troop that has knows_common will have every skill listed here; a Riding skill of 1, a Trade skill of 2, an Inventory Management skill of 2, a Prisoner Management skill of 1, and a Leadership skill of 1. knows_common is what is known as a constant ; a phrase that represents something else, be it a number, an identifier, another constant, or any other valid object. A constant can represent any number of objects, as long as those objects are in the right order for the place where you intend to use this constant.

In this case, knows_common is defined as having everything on the right side of the = sign. So in effect, by putting knows_common in the Skills field, the module system will function just as if you had typed out all of that in the skills field.

Now let us look at the next entry in the list.

```
["regular_fighter","Regular Fighter","Regular Fighters",tf_guarantee_boots|
    tf_guarantee_armor,no_scene,reserved,fac_commoners,
[itm_hide_boots],
str_8|agi_8|level(11),wp(90),knows_common|knows_ironflesh_1|knows_power_strike_1|
    knows_athletics_1|knows_riding_1|knows_shield_2,mercenary
_face_1, mercenary_face_2],
```

In this example, you can see the slightly stronger "regular fighter"; this one has higher ability scores, is level 11, and knows some skills beyond knows_common. In-game, if some "novice fighters" in our party had reached sufficient experience to reach level 11, we might want to allow them to upgrade into "regular fighters". We will see later how troop trees are done.

In a similar way you can declare constants for the attributes (including the level or not) and the weapon proficiencies, e.g. as follows:

```
def_attrib = str_7 | agi_5 | int_4 | cha_4
knight_attrib_1 = str_15|agi_14|int_8|cha_16|level(22)
wp_swadia_low_tier = wpex(90,75,75,50,50,60)
```

Again, take care that those objects are in the right order for the place where you intend to use new declared constants and make sure that you implement the constants properly into the troop tuple. Look up examples in the Module System of Native or of other Open Source Code Mods.

5.3.3.1 Troop Flags

tf_male determines that the character is male.

tf_female determines that the character is female.

tf_undead is basically the third gender/race (same thing in M&B). It can't be simply used since it requires some more work at module_skins and some bad skinning at the default undead meshes.

tf_hero is given to every unique NPC which the player can encounter in-game, even the merchants are counted as Heroes. The main differences between Heroes and regular troops are:

- Heroes are unique, you can have only one of each Hero unless they are cloned by error or by design. Even by design, however, cloning Heroes is a bad idea. Since there should be only one specimen of each Hero, they have no plural troop name. Field 3 of the Hero tuple is therefore identical to Field 2.

- Heroes are unkillable, their health is represented by a percentage value.
- Heroes take up a full party stack, each an own one.
- Heroes show up properly in a scene when they are assigned to one in their troop tuple.
- Heroes stay with the player when he is defeated by an enemy party - the Heroes are not captured by the enemy - but the player can capture enemy Heroes as normal.

tf_inactive is used for non-NPC troops like chests, markers, etc. Troops with this flag are not dressed even with other flags set. Normally scene props or meshes get spawned at their positions.

tf_unkillable deprecated, not useable anymore.

tf_allways_fall_dead lets hero and non-hero troops always die in battle, so they can't get knocked out and healed later.

tf_no_capture_alive is a deprecated flag from the old M&B that determined that the Troop can't be captured alive, it can't become a Prisoner. Not useable anymore.

tf_mounted enables that the Troop's movement speed on map is determined by riding skill.

tf_is_merchant causes Heroes to not equip any of the items in their inventory, except what they are originally assigned in their troop tuple. In other words, these merchants can receive all sorts of items over the course of the game, but they will not wear or use the items, and the items will properly show up for sale.

tf_randomize_face randomizes the face at the beginning of the game. Requires two face codes at the troop tuple.[123]

tf_guarantee_boots guarantees the troop will have boots if there are some in their loadout.

tf_guarantee_armor guarantees the troop will have an armour if there is one in their loadout.

tf_guarantee_helmet guarantees the troop will have a helmet if there is one in their loadout.

tf_guarantee_gloves guarantees the troop will have gloves if there are some in their loadout.

tf_guarantee_horse guarantees the troop will have a horse if there is one in their loadout, and puts the troop in the cavalry troop category (overrides the flag tf_guarantee_ranged).[124]

tf_guarantee_shield guarantees the troop will have a shield if there is one in their loadout.

[123] Waldzios, Modding Q&A.

[124] Dusk Voyager, Modding Q&A.

tf_guarantee_ranged guarantees the troop will have a ranged weapon if there is one in their loadout, and puts the troop in the archer troop category.

tf_guarantee_polearm guarantees the troop will have a polearm if there is one in their loadout.

tf_unmoveable_in_party_window means that unless the troop is an enemy prisoner, you cannot garrison this troop or trade it to another party. Normally used for Hero troops.

Two more flags are defined in module_troops.py which are combinations of above explained flags:

```
tf_guarantee_all = tf_guarantee_boots|tf_guarantee_armor|tf_guarantee_gloves|
    tf_guarantee_helmet|tf_guarantee_horse|tf_guarantee_shield|tf_guarantee_ranged
tf_guarantee_all_wo_ranged = tf_guarantee_boots|tf_guarantee_armor|
    tf_guarantee_gloves|tf_guarantee_helmet|tf_guarantee_horse|tf_guarantee_shield
```

5.3.3.2 Scene Entry Point

Tuple field five goes hand in hand with the mission template entry flag `mtef_scene_source`. This flag is used for troops that have a site/entry point defined statically. It spawns them on the first frame of the mission only if the entry point defined in the tuple has the flag.[125] Could be quasi deprecated due to the usage of other scripts, up4research.[126]

Work in other posts here[127]

two ways to spawn troop[128]

5.3.3.3 Troop Faction

troops can't have multiple factions, morale script[129]

5.3.3.4 Attributes

A character has four major attributes, which affect what you can do in very profound ways. Attributes do two things: they give you some kind of immediate advantage, and they also allow you to increase skills that are dependent on that attribute. Strength and Intelligence also allow better armor or books to be used, respectively. Skills can only be raised to one third of a certain attribute (rounded down). For example, Ironflesh is a Strength based skill, so a character with 14 Strength can only

[125] cmpxchg8b, Modding Q&A.

[126] Somebody, Modding Q&A.

[127] Lumos, Modding Q&A, MadVader, Modding Q&A, Lumos, Modding Q&A, Docm30, Modding Q&A and other note.

[128] kalarhan, Modding Q&A.

[129] Lav, Modding Q&A.

raise their Ironflesh up to level 4, while 15 Strength will allow Ironflesh up to level 5. Furthermore, each attribute carries its own benefits.

When players create a new character in Native after selecting a background, they will get four points to spend on attributes, and upon each level-up, they will receive one more. These choices cannot be changed once committed. The maximum level for attributes is 63.[130]

Strenght (STR) Every point adds +1 to hit points. (Hidden bonus: increases damage of melee weapons, bows, and thrown weapons).

Agility (AGI) Each point gives five weapon points and slightly increases movement speed on the battlefield.

Intelligence (INT) Every point to intelligence immediately gives one extra skill point.

Charisma (CHA) Each point increases your party size limit by +1.

[130] _Sebastian_, Modding Q&A, and Garedyr, Modding Q&A.

5.3.3.5 Random Attribute and Skill Points

Sometimes you will encounter in-game that after adding or editing troops in the Module System, their attributes (STR, AGI etc.) and/or skills will vary after compiling from the values which you actually assigned them. This has to do with the level and skill redistribution which occurs with every new game. The engine checks that every troop has the total amount of attribute and skill points which they should have accordingly to their assigned level. If there are less points distributed, the game assigns by itself randomly attribute and skill points. You can avoid this by either reducing the level or cranking up useless stats like Charisma or Persuasion.[131]

Alternatively you can of course distribute them accordingly to their level.[132] For attributes the formula is as follows:

```
Native:
Total Attribute Points = 17 + 16 + Troop Level * 1
In general:
Total Attribute Points = 17 + Defined Start Attribute Points + Troop Level *
    attribute_points_per_level (module.ini setting)
```

The defined start attribute points can be found in the first hard-coded tuple at module_troops.py, the player one, which cumulate to 16 in Native.

```
  ["player","Player","Player",tf_hero|tf_unmoveable_in_party_window,no_scene,
      reserved,fac_player_faction,
   [],
   str_4|agi_4|int_4|cha_4,wp(15),0,0
      x000000018000000136db6db6db6db6db00000000001db6db0000000000000000],
```

For the skill points take note that each point given to the attribute Intelligence gives one additional skill point. The formula is thus as follows:

```
Native:
Total Skill Points = 13 + Troop Level * 1 + INT
In general:
Total Skill Points = 13 + Troop Level * skill_points_per_level (module.ini setting
    ) + INT
```

At your character creation you can also keep the limiter in mind. In Native skills can only be raised to one third of a certain attribute (rounded down). You can alter this via a setting in the module.ini file of your mod, `attribute_required_per_skill_level`, which has a default value of 3 at Native.

[131] Somebody, Modding Q&A.

[132] Earendil: The following formulas are constructed by using the player troop in-game. They can thus contain misstakes.

spare skill points at higher levels of troops[133]

5.3.3.6 Health/Hit Points

health formula: 35 + skl_ironflesh * 2 + strenght

5.3.3.7 Proficiencies/Weapon Points

Weapon Proficiencies are the player's ability to use specific weapons. There are six in total and they include: One Handed Weapons, Two Handed Weapons, Polearms, item Archery, item Crossbows and item Throwing. In With Fire & Sword (WFaS) the crossbows are getting replaced by firearms. As you might have noticed there are also mods which have both since there is also a hidden firearm proficiency which can get activated. Both, the replacement and the activation, by setting the respective lines in module.ini to 1 instead of 0.

```
use_crossbow_as_firearm = 0
display_wp_firearms = 0
```

The three melee proficiencies covers the use of swords, axes, spears, maces, etc. Polearms can be one- or two-handed but are considered their own weapon class. Using a weapon will increase its related proficiency. Increasing a proficiencies allows the character to deal more damage with a melee weapon, and use ranged weapons more accurately. It also improves the weapon's attack speed.

The player can increase proficiencies in two ways: By simply using the weapon in combat and by spending Proficiency Points gained upon level-up on the proficiency of their choice. The maximum level for any weapon proficiency is 699, where points cannot be added to a weapon proficiency anymore. The Weapon Master skill level imposes a cap on all the proficiencies, restricting how these points can be spent. The cap does not prevent increases earned through actual use. Each point of Weapon Master increases the standard proficiency limit. The proficiency cap progression starts at 60 (for a Weapon Master skill of 0) and the cap increases by 40 points for each additional point of Weapon Master skill.

The higher the Proficiency is, the more Proficiency Points are needed to raise it further. For example, to increase the One-Handed Proficiency by one point when it is at 20 would take one Proficiency Point. If that skill was 380, the player might need to spend between 10 and 16 Proficiency Points to increase the proficiency by one point. This amount varies based on how high the Weapon Master skill is. The higher the proficiency skill, the more points one will have to spend.

Increasing weapon proficiencies will result in increased damage (when using a melee weapon or a ranged weapon if it benefits from Power Draw/Throw skill), faster attack speeds and increased

[133] TheCaitularity (credit), Modding Q&A.

accuracy (when using a ranged weapon). All results are gradual and may need as many as 50 points in a proficiency to see a notable difference, though results are much more noticeable while proficiency is still low. The Weapon Master skill increases the rate at which proficiency points are earned, and raises the spending cap on all proficiencies.

Proficiency increases more quickly when fighting higher level opponents and slower when fighting lower level opponents, in addition to other factors. It is also possible to benefit more from a single attack if the attack was particularly difficult or damaging. The more damage that is dealt with a weapon, the more experience the player earns towards that weapon's proficiency. Ranged weapon proficiencies increase faster when more difficult shots are landed rather than when damage is done (damage still counts, just not as much). Headshots provide a bonus to the difficulty multiplier as well as shooting while on a moving horse. Additionally, the type of ranged weapon used also affects shot difficulty. Thrown weapons receive a moderate increase in shot difficulty as they are universally shorter ranged than other ranged weapon types while crossbows receive a penalty for their ease of use. Bows receive neither a bonus or a penalty to shot difficulty calculations.

The weapon proficiency is not affected by random point redistribution as it happens with the attribute and skill points as mentioned above. You can still try to achieve a roughly fitting setting. Take note that each point given to the attribute Agility gives five additional weapon proficiency points. The formula you can orientate upon is thus as follows:

```
In general:
Total Weapon Proficiency Points = Troop Level * weapon_points_per_level (module.
    ini setting, 10 in Native) + 5 * AGI + Defined Start Weapon Proficiency Points
    + parameter of own choice between 10 and 60
```

Add little passage about how to edit the hard-coded proficiency names.[134] Need to look up where else such note could be useful (if not opening up a new capitel with new module files extension)

5.3.3.8 Creating Face Codes

If the troop you want to assign a facecode to is a "hero" (a lord, merchant or any other troop that's only supposed to turn up once), then you only need to set up the tuple field 12 (there are some exceptions, mostly ladies). For everyone else, the game chooses a random face between the two face codes in the tuple fields 12 and 13. Bear in mind that the game's idea of "randomisation" is incredibly screwy and without setting a dummy skin/race in module_skins.py, it will mostly ignore your choices.[135] To get rid of the problem of reused faces you need to add a dummy skin to module_skins.py since there is a hard-coded switch that disables that functionality completely if the number of skins is not 2.

[134] Specialist, Modding Q&A.

[135] jacobhinds and Ikaguia (upfollowing post), Modding Q&A.

The purpose of that functionality was probably to improve performance.[136] How the randomisation works exactly is unknown.[137]

With Edit Mode enabled, go to the Character menu and click the Edit Face button. Play around with all the sliders and settings until you have got the face you are after. Now click Ctrl+E, and you will notice a box/button pop up at the top right with a string of characters in it. Click this and it will copy the face code to the clipboard. In module_troops.py, simply paste in it in as the face code.[138] You can also make use of the predefined face codes given to you at the top of module_tropps.py, starting with swadian_face_younger_1, for example

```
woman_face_1 = 0x0000000000000001000000000000000000000000001c00000000000000000000
```

and add woman_face_1 as a valid face code into your troop tuple. You are free to create your own face code definitions at the top. Beneath the for Native predefined face codes you will also stumble above reused definitions like

```
refugee_face1 = woman_face_1
```

You will perhaps think 'Why not simply using woman_face_1 again?'. The reason is simply that you might want to give the refugees at a later stage of development more individual face codes but are for now setting another defined one as a kind of placeholder.

When you are starting a new race, add a new troop near the beginning of module_troops.py

[136] cmpxchg8b, Modding Q&A.
[137] Questionlog to K700
[138] Thorgrim, Modding Q&A.

(but after the hard-coded ones). Then go to the Character menu ingame and press “next“ above the image of the player until you get to the new race troop. Setting the initial face codes of such a new troop to zero before will save you time to set this template troop’s face codes directly.[139]

At the multiplayer you have to differentiate between troop templates for the players and troop tuples for the bots (Native differentiates them for example with an additionally _ai at the troop id). Multiplayer troop faces are overriden by the ones in the player’s profile engine-side, so the face codes which you assign are only functionable for the bots. What you can do, however, is spawn another agent and use the operation player_control_agent instead of player_spawn_new_agent.[140]

facecode seed determines troop loadout?[141] Perhaps also with reference to the later subchapter about this.

face key operations[142]

Facecode stuff[143]

MP hero troops need always the same face code twice to work normally[144]

setting facecode to 0[145]

5.3.3.9 Troop DNA

It might very well be that it’s a footstep subchapter to equipment selection.

troop dna[146]

troop dna[147]

5.3.3.10 Equipment Selection of Bots/AI Troops

bots with more than one weapon[148]

bots choosing armour[149]

bots choose items based on price (search before reasking K700) some posts later linked to another post which I should have also marked[150]

multiple item entries in troop inventory[151]

[139] jacobhinds (read up until post #19.723), Modding Q&A.
[140] Somebody, Modding Q&A.
[141] Somebody, Modding Q&A and Modding Q&A.
[142] kalarhan, Modding Q&A.
[143] SupaNinjaMan, WSE problem.
[144] TortenSkjold, Mount & Blade Modding Discord.
[145] jacobhinds, Modding Q&A.
[146] Fisheye, M&B Scripting Q&A.
[147] Lav, Lumos and cmpxchg8b, Modding Q&A, Modding Q&A and Modding Q&A.
[148] jacobhinds, Modding Q&A.
[149] kalarhan, Modding Q&A.
[150] kalarhan, Modding Q&A.
[151] kalarhan redirect, Modding Q&A.

item selection distribution[152]

ai at weapon selection[153]

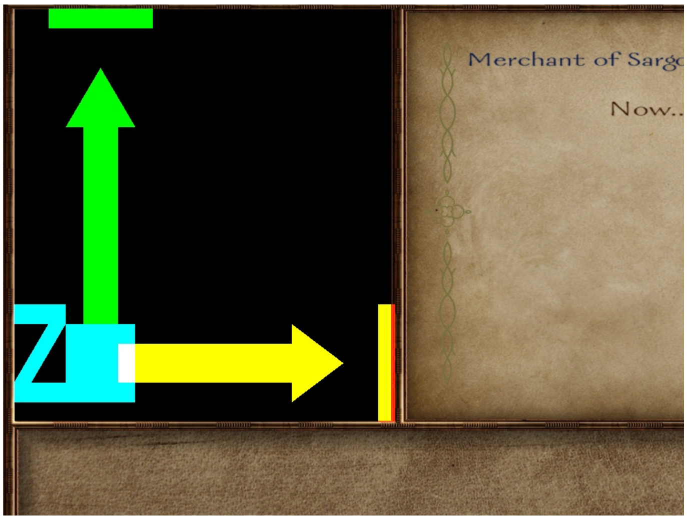
AI selecting armour[154]

item distribution of bots[155]

weapon guarantee [156]

5.3.3.11 Troop Image

During dialogue you normally see the current speaking troop inside a conversation scene, with camera movement between speakers. However by using the **Troop Image** entry, you can set a static image to be displayed instead. While it is recommended to use a 2D planar mesh, the entry will accept any mesh from any brf file loaded inside the module.ini. Unlike similar features in menus and presentations it is not necessary to register the mesh inside module_meshes.py, you only need to enter the mesh name from the brf as a string.



The mesh is rendered orthographically with the camera facing downwards from +Z to -Z with +Y being Up and +X being Right. The meshes are not required to be centered as they are centered automatically. Meshes are scaled to fit the boundaries of the dialogue window. The forced scaling can cause rendering issues such as squashing or stretching the mesh to fit the aspect ratio of the window. Furthermore, for non-rectangular meshes, the space behind the mesh isn't redrawn each frame, causing a visual artifact where the relations pop-up stays on screen.

Note that areas with partial transparency will become less and less transparent each frame as it is drawn over, and added to, the previous image of the mesh each frame. Total transparency will not

[152] kalarhan, Modding Q&A.

[153] InVain, kalarhan and _Sebastian_, Modding Q&A, Modding Q&A, Modding Q&A and Modding Q&A.

[154] jacobhinds, Modding Q&A, and kraggrim, Modding Q&A.

[155] kalarhan, Modding Q&A.

[156] Somebody, Modding Q&A.

have this problem.

Also of note, if two troops are talking, one with a Troop Image and one without, the camera will never show the second troop, just the Troop Image. Further testing is required to see what happens when multiple troops with images are in the same conversation, however it appears the first speaker will take priority and be featured as the image.

5.3.3.12 Troop Trees

The list of which troops can be upgraded into what is contained at the very bottom of module_troops. This defines which troops can be upgraded, including what they can upgrade into, when the experience conditions are met.

As you can see, each troop's upgrade choices must be defined here through the operation upgrade(troops). The first string is the ID of the troop to be upgraded, the second string is the ID of the resulting troop. For example, upgrade(troops,"farmer", "watchman") will allow a "farmer" to upgrade into a "watchman", when the "farmer" has accrued enough experience.

There are two types of upgrade operations:

- upgrade(troops,"source_troop", "target_troop_1") offers only one upgrade choice; "source_troop" to "target_troop_1".
- upgrade2(troops,"source_troop", "target_troop_1", "target_troop_2"), however, offers the player a choice to upgrade "source_troop" into either "target_troop_1" or "target_troop_2". Two is currently the maximum number of possible upgrade choices.

Notice that the upgrade section is outside of the main code block for troops (after the last ']'). It is important to watch for where aspects of the code are located, as it is important for the compiler to see things in a specific order. Also note that here they do not have commas (,) after each line. Always keep in mind where you are in or outside the code, and the conventions being used.

At the upgrade event there are two scripts getting called by the engine. When you select a troop in the party window, the game calls the script "game_get_upgrade_cost" once for each available upgrade path - that is, once or twice. You can check if the script is supposed to be called once or twice by checking that the troop has a second upgrade path available. When the troop is upgraded, the script "game_get_upgrade_cost" is called once to get the cost of selected upgrade, and then twice to get all upgrade costs. No idea why. So for troops with one upgrade path, the script is thus called three times during the upgrade and for troops with two upgrade paths five times.[157]

[157] Somebody, Modding Q&A, and Caba'drin and Lav, Modding Q&A. In case you want to change how the upgrade costs mechanic is working read Lav's comment en detail and also the post of Waldzios, [WB] Warband Script Enhancer v3.2.0.

Note that the troop tree can't have more than two upgrade paths. This is a hard-coded limit, not only in the party screen, even the `"Troop"` C++ class of the game engine has only two upgrade fields.[158][159] This is however just the normal system. There is nothing preventing you from creating your own system via own code, UI, etc. It is just super easy to use the two-path system as that one is already all done and ready for use, that is all.[160]

5.3.3.13 Differences between Classes, Subclasses and Divisions

Classes are from the original M&B. They are always - unless you code them differently - either infantry, archers, or cavalry (class numbers 0, 1, and 2), depending on the guarantee flag which you set for them (infantry is set on default, archers if `tf_guarantee_ranged` and cavalry if `tf_guarantee_horse` is set, with latter one overriding the former flag). Classes apply thus to troop templates, while divisions only apply to agents in battle. Divisions have been introduced with Warband and you can call them in a battle with the number keys (1-9 or division numbers 0-8). For the most part, you can ignore classes in Warband without much consequence. Sub-classes are a messy way to refer to divisions, typically, in the various order operations.[161]

5.3.3.14 Division/Class Settings of Troops

Some notes regarding the flags

division setting of troops[162]

5.3.3.15 Other Notes

Perhaps some little chapter about how to quasi revive the flag? no capture alive outdated[163] Are those troop flags cancelling each other? One shouldn't even work[164]

Make sure that those settings are all already present in latest Native MS. firearm proficiency[165]; top part module troops[166]

agent set x modifier[167]

[158] cmpxchg8b, Problem with upgradeoptions for one trooptype.

[159] While KON_Air demonstrated at the MBX forum, Random Notes, that it was possible at an older engine version to add a third troop path, it does not seem like he found a way to make use of it or to test if it is even working.

[160] kalarhan, Modding Q&A.

[161] Caba‘drin, Modding Q&A.

[162] Lav, Modding Q&A.

[163] NPC99, Modding Q&A and Modding Q&A.

[164] Discussion, Modding Q&A.

[165] Somebody, Modding Q&A.

[166] Somebody, Modding Q&A.

[167] Somebody, Modding Q&A.

5.3.4 Module Items

```
Each item record contains the following fields:
1) Item id: used for referencing items in other files.
   The prefix itm_ is automatically added before each item id.
2) Item name. Name of item as it'll appear in inventory window
3) List of meshes.  Each mesh record is a tuple containing the following fields:
  3.1) Mesh name.
  3.2) Modifier bits that this mesh matches.
   Note that the first mesh record is the default.
4) Item flags. See header_items.py for a list of available flags.
5) Item capabilities. Used for which animations this item is used with. See
    header_items.py for a list of available flags.
6) Item value.
7) Item stats: Bitwise-or of various stats about the item such as:
   weight, abundance, difficulty, head_armor, body_armor,leg_armor, etc...
8) Modifier bits: Modifiers that can be applied to this item.
9) [Optional] Triggers: List of simple triggers to be associated with the item.
10) [Optional] Factions: List of factions that item can be found as merchandise.
```

module_items.py begins with a number of constants that are used to govern item modifiers which adjust the items we see in-game. Bent polearms, chipped swords, heavy axes, they are all created from a tuple in module_items and then given an appropriate item modifier to adjust their stats up or down.

After the constants, the list of tuples begins. The first one of interest is the weapon "practice_sword", which will function as a good example. Let's look at this tuple:

```
["practice_sword","Practice Sword", [("practice_sword",0)],
itp_type_one_handed_wpn|itp_primary|itp_secondary|itp_wooden_parry|itp_wooden_attack, itc_longsword,
3,weight(1.5)|spd_rtng(103)|weapon_length(90)|swing_damage(16,blunt)|thrust_damage(10,blunt),imodbits_none],
```

This is a basic practice weapon, used in the arena fights and training fields. Practice_sword tuple examination:

```
1) Item id = "practice_sword"
2) Item name = "practice_sword"
3) List of meshes:
  3.1) Mesh name = "practice_sword"
  3.2) Modifier bits = 0
4) Item flags = itp_type_one_handed_wpn|itp_primary|itp_secondary|itp_wooden_parry|itp_wooden_attack
5) Item capabilities = itc_longsword
6) Item value = 3
7) Item stats = weight(1.5)|spd_rtng(103)|weapon_length(90)|swing_damage(16,blunt)|thrust_damage(10,blunt)
8) Modifier bits = imodbits_none
9) Triggers = None.
```

We can now tell all of "practice_sword"'s details from its tuple:

- It uses the mesh "practice_sword" as its default mesh.
- It uses item flags that mark it as a one-handed melee weapon. Troops who are equipped with "practice_sword" will consider "practice_sword" when choosing a primary melee weapon from their inventory listing. They will also consider "practice_sword" when choosing a backup (secondary) weapon. (Note: The secondary weapon function is currently nebulous. It's unclear when it's used or if it's used at all. However, melee troops certainly will not switch to different melee weapons during combat.)
- It has all the animations defined in the constant itc_longsword, so "practice_sword" is able to be used as a long sword would be.
- It weights 1.5 kilograms. Its speed rating is 103, its weapon length is 90. It does a base 16 blunt damage while swinging, and a base 10 blunt damage while thrusting.
- It uses no item modifiers.

5.3.4.1 Item Types/Classes

There exists a range of item types which you should not confuse with item properties even if they have the same shorttag in front of their names.[168] The weapon item types decide which animations are getting used for the idle stance and movements of the agent carrying them.[169] The armoury item types on the other hand replace specific parts of the body meshes of the agent. The remaining item types are horses, animals, goods and books. Keep in mind that an item is not intended to have more than one item type. Also, you cannot declare a new weapon, armour or horse item type, you can only repurpose the Native ones, which may or may not have the correct set of animations and AI behaviour associated with it.[170].

itp_type_one_handed_wpn	itp_type_bow	itp_type_arrows
itp_type_two_handed_wpn	itp_type_crossbow	itp_type_bolts
itp_type_polearm	itp_type_thrown	itp_type_bullets
itp_type_shield	itp_type_pistol	
	itp_type_musket	
itp_type_head_armor	itp_type_horse	itp_type_goods
itp_type_body_armor	itp_type_animal	itp_type_book
itp_type_foot_armor		
itp_type_hand_armor		

While the weapon and armoury item types as well as the item class horse are pretty self-explanatory, the three rather unusual item classes animals, goods and books might irritate you and shall get explained shortly:

itp_type_animal Animals are basically the same as Horses regarding the animations and the item entry. The difference is that agents with this flag instead of itp_type_horse can't be mounted by human agents (the dialog box to mount does not appear) and the item can appearantly also not be used at the food slot (see Subchapter 5.3.4.15 for the details) in case you have reactivated that one.

itp_type_book Books serve no purpose in the M&B game series except for being items to raise the player's character stats. They can either appear in form of readable or reference books. Readable books are one time use objects giving the player an one-time skill or attribute bonus after the book has been read completely. Reference books on the other hand give the player

[168] This is especially important at the usage of the two operations `item_get_type` and `item_has_property`.

[169] In contrast to them the item capability flags are used for attack, defend and carry animations. To see which item capability and item property type flags are connected with which animations and animation sequences see the chapters 5.3.5.9 and 5.3.5.10.

[170] Somebody, Modding Q&A. Other item types could possible be added and will need to get their effects scripted in manually, up4research.

a skill bonus while he has the book in his inventory. At both kind of books you need to take care that you have placed new books alongside the others, respecting the declared constants in module_constants.py.

itp_type_goods Goods summarise all food and manufactured items, excluding armoury, which are getting sold by merchants in the villages or towns or looted when raiding a caravan or a bandit lair. They are either needed for provision or for trade purposes and are also used a couple of times as quest items. Goods affect with their weight the travel speed at the world map.

5.3.4.2 Item Properties

There exists a list of item property flags which assigns specific properties to each item. You have to differentiate here between item property type flags, which specify the type of the item as you can find them at the section about the item stats, and more genuine item property flags as they are listed below, in alphabetic order:

itp_always_loot ensures that the item will always appear in the normal post-battle loot screen.

itp_attach_armature needed for rigged items.(?)

itp_bonus_against_shield multiplies the damage by 2.0 when striking vs shields and by 1.1 vs destructible constructions.[171]

itp_can_penetrate_shield enables the item to penetrate shields. It is only for ammunition.

itp_can_knock_down gives the weapon a chance to knock down the enemy.

itp_cant_reload_on_horseback prevents reloading the weapon while being on horseback.

itp_cant_reload_while_moving forces the player to stand still for reloading the weapon.

itp_cant_reload_while_moving_mounted forces the player to stand still with his horse for reloading the weapon.

itp_cant_use_on_horseback disables the item to be used on horseback.

itp_civilian marks civilian cloths and/or armour which can be worn when indoors.

itp_consumable marks consumable items which perish slowly with each day. Used in Native Module System in an inconsistent way.

[171] _Sebastian_, Modding Q&A.

itp_couchable makes spear/lance couchable, couched lance damage will be applied at successful attacks.

itp_covers_beard removes beard mesh for armours only.

itp_covers_head removes head mesh.

itp_covers_hair removes hair mesh for armours only.

itp_covers_hair_partially is a flag which has been introduced with VC, meant to fix the clipping issue from the flag before. For this it is using a vertex animation for the hair meshes and causes that one to switch to its alternate frame if it has one. Frame 1 needs the setting 10 for this, see the hairs at VC as examples.[172]

itp_covers_legs removes the leg meshes of the body. If you do not have this flag on a piece of body armor, it will load `def_trousers` as a submesh of the armor.[173] It's a purple default mesh which you can find in item_meshes1.brf.[174]

itp_crush_through affects that the melee weapon can break through blocks, dealing damage and stuns over a shield or weapon block. It does not happen always though, it depends on the speed bonus, the resistance of the shield, and some luck. It tends to occur against a shield in most cases, and very rarely against weapons. When on foot, only the overhead swing can result in a crushing blow. Not sure yet how crush_through_treshold = 2.4 in module.ini affects it, up for research.

itp_default_ammo sets this item as default ammo, like a specific bolt for all crossbows. Without an additional ammo pack, the preloaded bolt will be the default ammo.

itp_disable_agent_sounds disable agent related sounds, but not voices; useful for animals.

itp_doesnt_cover_hair lets the item not cover the hair. Clipping will probably occur, not sure what the difference is to not having this flag set at all.

itp_extra_penetration is - based on the given informations - a flag introduced with WFaS which adds a multiplicator to the attack damage. The multiplicators can be set in module.ini, the

[172] kalarhan, Modding Q&A, and Fredelios, Modding Q&A.

[173] Somebody, Modding Q&A, and dstn, Mount & Blade Modding Discord. Somebody and Mandible talk here about a tag called `doesn't_cover_legs`, it's however literally an armour without `itp_covers_legs`.

[174] Somebody, Modding Q&A.

respective lines are extra_penetration_factor_soak = 1.0 and extra_penetration_factor_reduction = 1.0. In Native there are no extra penetration flags set, so they are kept ineffective.[175]

itp_fit_to_head should deform the helmet item such way that it adapts to the form of the head mesh. Up for research.

itp_food lets the item give a morale bonus to the party. Used in Native Module System in an inconsistent way.

itp_force_attach_left_forearm forces the item to be always carried at the left forearm.

itp_force_attach_left_hand forces the item to be always carried with the left hand.

itp_force_attach_right_hand forces the item to be always carried with the right hand.

itp_force_show_body forces showing body (works on body armour items).

itp_force_show_left_hand forces showing left hand (works on hand armour items).

itp_force_show_right_hand forces showing right hand (works on hand armour items).

itp_has_bayonet adds a special agent animation when switching weapon modes, it still needs the flag itp_next_item_as_melee to have a secondary item as melee. Without the flag weapon mode switching happens instantly without any animation.[176]

itp_has_upper_stab lets weapons stab instead of swing at upper attacks.

itp_ignore_friction prevents the item to get slowed down by the air.

itp_ignore_gravity prevents that trajectory bends the item downwards, so it becomes weightless. Disables effectively a bullet drop.

itp_is_pike allows for a new attack animation when crouching, pike bracing.

itp_merchandise makes an item available in shops.

itp_next_item_as_melee lets the weapon use the next item as melee weapon, it has to have the same mesh though. You can only toggle `ranged->melee` or `melee->melee`.[177]

[175] The flag will not work on ammunition, probably caused by the same logic that takes the firing weapon's damage type instead of the ammo's; ithilienranger and MadocComadrin, Modding Q&A. Not sure if this flag can only be added to melee weapons or to missile weapons too. Specialist hints in a message that he believes it also work on shields but he never posted any test results. Therefore up for research.

[176] _Sebastian_, Modding Q&A.

[177] I still need to understand the bottom part of this discussion here.

itp_no_blur removes the blur effect when swinging a spear.

itp_no_parry disables the parry function of the item, typically given to weapons which are short, with a reach no greater than 60.

itp_no_pick_up_from_ground prevents that the player can pick up this item from the ground.

itp_offset_lance moves the polearm forward, holding it further down the shaft, for couching purposes.

itp_offset_musket flips item upside down, offset -20 + weapon_length

itp_penalty_with_shield gives a damage penalty when used with a shield.

itp_primary is marking a weapon such way that Troops who are equipped with it will consider it when choosing a primary melee weapon from their inventory listing.

itp_remove_item_on_use is for ammo like bolts, arrows and throwables. After a battle ends they won't refill normally. The operation `agent_refill_ammo` won't refill it neither.[178]

itp_secondary is marking a weapon such way that Troops who are equipped with it will consider it when choosing backup (secondary) weapon. (Note: The secondary weapon function is currently nebulous. It's unclear when it's used or if it's used at all. However, melee troops certainly will not switch to different melee weapons during combat.)

itp_two_handed lets the player carry only this item actively, so it is not possible to carry additionally a shield type item.

itp_unbalanced enables that the weapon can't cancel hits mid-swing and also delays blocking when an attack has been finished.

itp_unique ensures that the item cannot be looted via the normal post-battle loot screen.

itp_wooden_attack gives the weapon a wooden sound at attacking.

itp_wooden_parry gives the weapon a wooden sound at parrying.

[178] Dalion, Mount & Blade Modding Discord.

5.3.4.3 Item Capabilities (itc & itcf)[179]

itcf's are item capability flags (basically what moves you can make with the item)[180] and **itc**'s are item capabilities, a combination of item capability flags and/or other item capabilities. So basically, taking the Bastard Sword as an example, all the itc_bastardsword is actually doing is combining two-handed and one-handed capabilities. Here is the entry in module_items.py of the weapon:

```
["bastard_sword_a", "Bastard Sword", [("bastard_sword_a",0),("bastard_sword_a_scabbard", ixmesh_carry)], itp_type_two_handed_wpn|itp_merchandise| itp_primary, itc_bastardsword|itcf_carry_sword_left_hip|itcf_show_holster_when_drawn,
 294 , weight(2.0)|difficulty(9)|spd_rtng(98) | weapon_length(101)|swing_damage(35 , cut) | thrust_damage(26 ,  pierce),imodbits_sword_high ],
```

Taken from header_items.py:

```
itc_bastardsword = itc_cut_two_handed |  itcf_thrust_twohanded | itc_parry_two_handed |itc_dagger
```

So itc_bastardsword is a combination of a few other combined capabilities and a non-combined capability (the itcf_thrust_twohanded).

Well, itc_bastardsword is obviously no good if you don't want the weapon to have a thrust attack. Instead you want one without the itcf_trust_twohanded. Oh, itc_nodachi has that:

```
itc_nodachi = itc_cut_two_handed | itc_parry_two_handed
```

Good. But what about the itc_dagger thing then?

```
itc_dagger = itc_cleaver | itcf_thrust_onehanded
```

Ah, so all you actually need is itc_nodachi and itc_cleaver. Hmm...

```
itc_morningstar = itc_cut_two_handed |  itc_parry_two_handed |itc_cleaver
```

There's your itc. Of course, I could just have said "it's itc_morningstar", but then you wouldn't have learned anything from it. Basic rule, always check the appropriate header files for the details when dealing with stuff like that or read the explanations at the beginning of the file, ike in this case the one of field five in the item tuple:

```
5) Item capabilities. Used for which animations this item is used with. See header_items.py for a list of available flags.
```

[179] Working in the informations given in the thread *Turning most one-handed swords into bastard type weapons* by Beaver, with additional other remarks.

[180] These flags are used for attack, defend and carry animations. It are the item property type flags that decides what animation is used on idle stance and for movement. To see which item capability and item property type flags are connected with which animations and animation sequences see the chapters 5.3.5.9 and 5.3.5.10.

You can of course simply define some new combined flags which will make adding these flags to weapons a lot easier.[181] Take however note that you cannot add new itcf flags, they are hard-coded into the game engine.[182] It is also recommended to check first for the already existing flags and look up their usage at the Native items before creating own new definitions. At the M&B game engine you have the following item capability flags available for weapon attacks:

itcf_thrust_onehanded	itcf_thrust_twohanded	itcf_thrust_polearm
itcf_overswing_onehanded	itcf_overswing_twohanded	itcf_overswing_polearm
itcf_slashright_onehanded	itcf_slashright_twohanded	itcf_slashright_polearm
itcf_slashleft_onehanded	itcf_slashleft_twohanded	itcf_slashleft_polearm
itcf_shoot_bow	itcf_throw_stone	itcf_reload_pistol
itcf_shoot_crossbow	itcf_throw_knife	itcf_reload_musket
itcf_shoot_pistol	itcf_throw_axe	
itcf_shoot_musket	itcf_throw_javelin	
itcf_shoot_javelin		
itcf_horseback_slashright_onehanded	itcf_overswing_spear	
itcf_horseback_slashleft_onehanded	itcf_overswing_musket	
itcf_thrust_onehanded_lance	itcf_thrust_musket	
itcf_horseback_slash_polearm		

Take note that itcf_throw_axe and itcf_throw_knife will also spin the item (with the knife letting the mesh stuck in body the opposite side the axes would), while itcf_throw_stone won't leave a projectile behind when it hits something.[183]

The flags `itcf_horseback_thrust_onehanded`, `itcf_horseback_overswing_right_onehanded`, `itcf_horseback_overswing_left_onehanded` and `itcf_thrust_onehanded_lance_horseback` are not getting used by the game engine and are thus deprecated.

Beside the above mentioned flags you have also the following item capability flags available for defend actions:

itcf_parry_forward_onehanded	itcf_parry_forward_twohanded	itcf_parry_forward_polearm
itcf_parry_up_onehanded	itcf_parry_up_twohanded	itcf_parry_up_polearm
itcf_parry_right_onehanded	itcf_parry_right_twohanded	itcf_parry_right_polearm
itcf_parry_left_onehanded	itcf_parry_left_twohanded	itcf_parry_left_polearm

Notice that an item does not necessarily need item capability flags which assign attack and defend actions to it. By giving an item only parry flags you will get a weapon with which you can only defend but not attack with.[184] Vice versa you can also create a weapon with which the player is not capable to parry. In a similar way you can create a weapon only useable by riders, by assigning

[181] This kind of flags which can be defined by yourself via combining other flags are so called utility flags.

[182] kalarhan, Modding Q&A.

[183] Somebody, Making An Item Throwable, and Dalion, Mount & Blade Modding Discord.

[184] GetAssista, Modding Q&A.

merely the horseback-only (and perhaps normal parrying) capability flags to it, although you should be aware of that the AI will still try to use their weapons in this fashion while dismounted.[185]

At last but not least there are the item capability flags with which you can specify for each weapon the *carry location* of its sheathed version. In contrast to the others before the game engine does not connected them with an animation but uses a set of hardwired positions and bones for the various wearable scabbards, quivers, etc.. So for each carry-locations the games has a fixed bone and a fixed roto-translation to apply.[186] The following ones are available to you:

itcf_carry_sword_left_hip	itcf_carry_revolver_right	itcf_carry_buckler_left
itcf_carry_axe_left_hip	itcf_carry_pistol_front_left	itcf_carry_crossbow_back
itcf_carry_dagger_front_left	itcf_carry_bowcase_left	itcf_carry_bow_back
itcf_carry_dagger_front_right	itcf_carry_mace_left_hip	itcf_carry_spear
itcf_carry_quiver_front_right	itcf_carry_axe_back	itcf_carry_board_shield
itcf_carry_quiver_back_right	itcf_carry_sword_back	itcf_carry_katana
itcf_carry_quiver_right_vertical	itcf_carry_kite_shield	itcf_carry_wakizashi
itcf_carry_quiver_back	itcf_carry_round_shield	itcf_show_holster_when_drawn

[185] Dusk Voyager, Modding Q&A, Somebody, Modding Q&A, and _Sebastian_, Modding Q&A.

[186] mtarini, How to mount scabbards in custom skins (was: bones). When using itcf_carry_quiver_front_right the mesh gets appearantly flipped/mirrored, _Sebastian_, Modding Q&A.

5.3.4.4 Item Stats

In this segment you will find a comprehensive list of stats[187] and a breakdown of their function. As some stats mean different things for different item types, I have organized the list by item types.

Generally:

abundance – Percentage value: This stat governs how often the item will appear in merchant inventories and combat loot. 100 is standard; can be more or less than 100 (down to 0), as 100 does not mean that it will show up 100 % of the time. Currently I do not know the ceiling number that equals 100 %.

weight – Kilogramme value: Defines the weight of the item in kilogrammes. The item weight limit of the Native Module System is 63.75 kg with a precision of up to 0.1 kg. Expanded Module Systems can have a higher weight limit up to 655.35 kg with a precision of up to 0.01 kg. Note that the game engine will round displayed weight to nearest 0.1 kg anyway.

itp_type_horse

body_armor – Value: Determines the horse's armour rating and number of hit points. A higher value means more armour and more hit points.

difficulty – Value: Determines how high the player's Riding skill needs to be to be able to mount this horse.

horse_speed – Value: The horse's speed on the battle map. Higher values make faster horses.

horse_maneuver – Value: The horse's manoeuvrability on the battle map.

horse_charge – Value: Determines how much damage the horse will do when charging infantry, and how much speed the horse will lose during each collision with an infantryman. Higher values will allow horses to do more damage and wade through more infantry.

itp_type_one_handed_wpn

difficulty – Value: The minimum STR score needed to be able to use this weapon. If a troop does not have greater or equal STR, he will not be able to wield it.

spd_rtng – Value: The attack speed of the weapon, both swing and thrust.

weapon_length – Centimetre value: The length of the weapon in centimetres. This stat determines how far the weapon will be able to reach in-game, regardless of the mesh size.

[187] For spd_rtng max is 255, higher values are truncated. Also 255 is max for all except horse_scale(max 1023), hit_points(max 65535), weapon_length(max 1023) and shoot_speed(max 1023); Wampirzy Lord, Rapid Fire.

swing_damage – Value, damage type: The base damage and damage type of the weapon when performing a swing attack.

thrust_damage – Value, damage type: The base damage and damage type of the weapon when performing a thrust attack.

itp_type_two_handed_wpn

Same as itp_type_one_handed_wpn.

itp_type_polearm

Same as itp_type_one_handed_wpn.

itp_type_arrows

weapon_length – Centimetre value: The length of the arrow in centimetres. Modifies how deep the arrow goes into the target (so bolts for example don't go all the way in on impact)

thrust_damage – Value, damage type: The amount of damage this type of arrow adds to the bow's base damage, and the damage type.

max_ammo – Value: The number of arrows in one stack. The maximum ammo is 255.[188]

itp_type_bolts

Same as itp_type_arrows.

itp_type_shield

difficulty – Value: The minimum Shield skill score needed to be able to use this shield. If a troop does not have greater or equal Shield skill, he will not be able to wield it. You need to add `difficulty(X)` alongside the other stats for it to appear since it's not implemented at Native items.[189]

hit_points – Value: The base number of hit points for this shield.

body_armor – Value: The amount of damage subtracted from every hit to the shield.

spd_rtng – Value: The speed with which the shield can be brought up into defensive mode.

[188] Somebody, Modding Q&A.

[189] GetAssista, Modding Q&A.

weapon_length – Value: Shield coverage, shield width. Higher values allow the shield to cover more body area, offering shield protection from arrows to larger sections of the body. Take note that it's the radius or half-width of the shield.[190]

shoot_speed – Value: Shield coverage, shield height. Needed for rectangular shaped shields to combine height with width. Use width only for round shields.[191]

itp_type_bow

difficulty – Value: The minimum Power Draw score needed to be able to use this bow. If a troop does not have greater or equal Power Draw, he will not be able to wield it.

spd_rtng – Value: The bow's reloading speed. IE, how fast a troop will be able to take an arrow from quiver, notch, and draw the bow again. Higher values will mean quicker reload times.

shoot_speed – Value: The speed at which ammunition from this weapon flies through the air. Higher values will mean faster ammunition; note, however, that very fast ammunition may clip through nearby enemies without hitting them, so try to keep it below 200m/s.[192] High shot speed will mean the projectile has less drop and so goes further.

thrust_damage – Value, damage type: The base damage and damage type inflicted by hits from this weapon.

accuracy – Percentage value: The chance of a shot flying exactly at the troop's aiming point. 100 represents a 100 % chance, lower values will greatly decrease the chance of a hit. This is not used on the Native bows and crossbows, but can be added.

itp_type_crossbow

difficulty – Value: The minimum STR score needed to be able to use this crossbow. If a troop does not have greater or equal STR, he will not be able to use it.

spd_rtng – Value: The crossbow's reloading speed. IE, how fast a troop will be able to take a bolt from quiver, notch, and pull back the string for firing. Higher values will mean quicker reload times.

[190] kalarhan, Modding Q&A.

[191] kalarhan, Modding Q&A.

[192] This is an old game engine bug that occurs with high missile speed, the maximum is 212.5 m/s before overflowing happens. The limit at the game engine is actually at 255 m/s. However, you have to differ if you spawn missiles via the operation `add_missile` or if a missile is shot by an agent. At latter case the game engine multiples the shoot_speed by 1.2 in the end, resulting in the 212.5 m/s limit. _Sebastian_, Mount & Blade Modding Discord. At the multiplayer the operation `add_missile` has a limit of 204.7 m/s since the value which is being sent over the network is capped to 2^11, so 0-2047 in 0.1 meter steps._Sebastian_, Mount & Blade Modding Discord

shoot_speed – Value: Same as at itp_type_bow.

thrust_damage – Value, damage type: Same as at itp_type_bow.

max_ammo – Value: The number of bolts that can be fired from this crossbow before it must be reloaded. The maximum ammo is 255.[193] [194]

accuracy – Percentage value: Same as at itp_type_bow.

itp_type_thrown

difficulty – Value: The minimum Power Throw score needed to be able to use this weapon. If a troop does not have greater or equal Power Throw, he will not be able to use it.

spd_rtng – Value: The reloading rate for this weapon. IE, how fast the next piece of ammunition can be readied for throwing.

shoot_speed – Value: The speed at which this weapon's ammunition flies through the air.

thrust_damage – Value, damage type: The base damage and damage type inflicted by hits from this weapon.

max_ammo – Value: The number of weapons (ammunition) contained in one stack. The maximum ammo is 255.[195]

weapon_length – Centimetre value: The weapon's length in centimetres.

itp_type_goods

food_quality The impact a food item will have on party morale. Values above 50 will improve morale while the item is being consumed, lower values will lower morale.

max_ammo The number of consumable parts for this item.

itp_type_head_armor

head_armor – Value: The amount of damage this piece of armour will prevent to the head of the troop.

body_armor – Value: The amount of damage this piece of armour will prevent to the body of the troop.

[193] Somebody, Modding Q&A.

[194] Be aware that AI agents won't use crossbows if they have not been given ammo, the same goes for pistols and muskets; MadVader, Modding Q&A.

[195] Somebody, Modding Q&A.

leg_armor – Value: The amount of damage this piece of armour will prevent to the legs of the troop.

difficulty – Value: The minimum STR required to wear this piece of armour.

itp_type_body_armor

Same as itp_type_head_armor.

itp_type_foot_armor

Same as itp_type_head_armor.

itp_type_hand_armor

Same as itp_type_head_armor.

itp_type_pistol

difficulty – Value: Warband does not specify the attribute/skill requirement for firearms. If you set the value higher than 0 you will make the weapon unuseable.

spd_rtng – Value: The pistol's reloading speed. IE, how fast a troop will be able to reload the pistol and aim it again.

shoot_speed – Value: The speed at which ammunition from this pistol flies through the air.

thrust_damage – Value, damage type: The base damage and damage type inflicted by hits from this pistol.

max_ammo – Value: The number of bullets that can be fired from this pistol before it must be reloaded. The maximum ammo is 255.[196]

accuracy – Percentage value: The chance of a shot flying exactly at the troop's aiming point. 100 represents a 100 % chance, lower values will greatly decrease the chance of a hit.

itp_type_musket

Same as itp_type_pistol. Keep in mind that pistols and muskets share bullets and firearm proficiency which is hard-coded into the game engine.[197]

itp_type_bullets

[196] Somebody, Modding Q&A.

[197] Somebody, Modding Q&A.

Same as itp_type_arrows. Note that you always need to tag a mesh as ixmesh inventory for bullets, so a ("cartridge_a", ixmesh_inventory) instead of the more commonly-styled ("cartridge_a", 0), or you get a niche bug with the bullets leaving their flying mesh in the scenery.[198] Add some note about hard-coded mesh[199]

Difficulty Requirement

You might ask yourself how to change on which Attribute or Skill a requirement for an armoury or a weapon is based on. However, you can't change this, it is hard-coded. You are also not able to introduce new requirements at other weapon types.[200]

[198] dstn, Mount & Blade Modding Discord.
[199] Hatonastick and Specialist, Modding Q&A.
[200] Lumos, Modding Q&A, and _Sebastian_, Moding Q&A.

Attribute Requirements in Native for Orientation

Armour either has no attribute requirement or Strength requirements between 6 and 9 (Plate Armour). The Black Armour has a requirement of 10. Heavy helmets such as the Winged Helmet also have a requirement of 10.

Melee weapons will require between 8 (Scimitar) to 12 (Sledgehammer) Strength, but certain weapon modifiers can add 1, 2, or 4 to Strength required. Thrown and Bow weapons indirectly require minimum Strength, due to their Skill requirements. Thrown weapons need 0-4 Power Throw before modifiers and Bows 0-4 Power draw before modifiers. Some Crossbows have strength requirements, those are in the 8-10 Range (10 for the Siege Crossbow).

Horses indirectly require Agility, as they require Riding skill. Armored horses require a minimum of 12 Agility (4 Riding). The modifiers Stubborn or Champion also increase skill requirements; +1 and +2 Riding respectively.

Books have Intelligence requirements between 7-12. In contrast to the other items the requirement for books is not getting set up in the item entry but in the script `initialize_item_info`.

5.3.4.5 Item Damage Types

As we've observed in the tuple examination, "practice_sword" does blunt damage. Obviously this is because it's a practice weapon, but it provides a good incentive to look into the different damage types available in M&B, to which weapons and ammunitions can get assigned to.

First, there is **cut damage**. Cut damage represents the slicing action of a sharp blade, such as a sword or an axe. Cut damage gets a bonus against unarmoured or lightly-armoured enemies, but conversely it has a large penalty against heavy armour. Cut damage will kill an enemy if it brings the enemy to 0 hit points.

Next we have **blunt damage**. Blunt damage represents the bludgeoning effect of weapons without an edge, such as a mace or hammer. Blunt damage gets a 50 % bonus against heavy armour, but blunt weapons are often shorter than cutting weapons and they do less damage overall. The largest advantage of blunt damage is that it knocks an enemy unconscious when the enemy is brought to 0 hit points, instead of killing him. Unconscious enemies can be captured and sold into slavery. There does even exist a special weapon order which tells troops to use blunt weapons. Charging horses also do blunt damage.

Finally, **pierce damage** represents the penetrating tip of arrows, crossbow bolts and similar weapons. Pierce damage gets a 50 % bonus against heavy armour, but piercing weapons usually do less damage overall in order to balance it with the other damage types. Pierce damage will kill an enemy if it brings the enemy to 0 hit points.

You can also influence how each damage type incluenfes the soak factor and the reduction factor of armour. Look up the settings `armor_soak_factor_against_*` and `armor_reduction_factor_against_*` at the module.ini documentation chapter.

5.3.4.6 Maximum Weapon Damage

As the ibf_damage_mask is 0x3ff and 2 bits are for damage type the maximum damage is 255. The Module System does damage & 0xFF to avoid the value leaking into the damage type bits, so that a 9999 damage is effectively cut down to 15 damage.[201]

```
0010 0111 0000 1111 (9999)
&
0000 0000 1111 1111 (0xFF)
=
0000 0000 0000 1111 (15)
```

5.3.4.7 Item Modifiers

Modifiers are labels that precede the name of an Armor, Shield, Weapon, or Horse, indicating either specific bonuses or penalties to defense, damage, skill, attribute requirements, speed ratings and/or costs. Every item has the modifier `plain` which is the normal default item.[202] The available modifiers are all listed in header_item_modifiers.py but also in the tables below which also give informations about their boni or penalties as well as their cost effects.

Take care that you differentiate between imods and imodbits and use them accordingly. The imodbits are the possible imods of the item. While an item entry gets assigned an imodbit (or multiple ones) at tuple field 8 of its item entry, it are at most (if not all) operations and scripts the imods for which you need to check for.[203] You add the possibility of an imod to be used or present on a item with the imodbit variable at the before mentioned tuple field on module_items.py. If you add a new sword you could have an entry similar to this one:[204]

```
["item", "Item", [("mesh",0)], ... , imodbits_sword],
```

Here the constant imodbits_sword gets used which is not one of the base modifiers but a predefined constant which you can find at the beginning of the file module_items.py. These defined constants consist of bundles of the standard item modifiers. You can of course still use only specific modifiers or decide to have none (using imodbits_none) at your item or define your own new constant, bundling

[201] dunde and cmpxchg8b, Maximum Weapon Damage?.

[202] Ikaguia, Warband Script Enhancer v3.2.0.

[203] kalarhan, Modding Q&A.

[204] The example shows a simple item entry without any item trigger or faction tuple field.

anew some of the standard item modifiers. The following constants for modifiers have already been set for ease of use:

```
imodbits_none = 0
imodbits_horse_basic = imodbit_swaybacked|imodbit_lame|imodbit_spirited|imodbit_heavy|imodbit_stubborn
imodbits_cloth = imodbit_tattered|imodbit_ragged|imodbit_sturdy|imodbit_thick|imodbit_hardened
imodbits_armor = imodbit_rusty|imodbit_battered|imodbit_crude|imodbit_thick|imodbit_reinforced|imodbit_lordly
imodbits_plate = imodbit_cracked|imodbit_rusty|imodbit_battered|imodbit_crude|imodbit_thick|imodbit_reinforced|imodbit_lordly
imodbits_polearm = imodbit_cracked|imodbit_bent|imodbit_balanced
imodbits_shield = imodbit_cracked|imodbit_battered|imodbit_thick|imodbit_reinforced
imodbits_sword = imodbit_rusty|imodbit_chipped|imodbit_balanced|imodbit_tempered
imodbits_sword_high = imodbit_rusty|imodbit_chipped|imodbit_balanced|imodbit_tempered|imodbit_masterwork
imodbits_axe = imodbit_rusty|imodbit_chipped|imodbit_heavy
imodbits_mace = imodbit_rusty|imodbit_chipped|imodbit_heavy
imodbits_pick = imodbit_rusty|imodbit_chipped|imodbit_balanced|imodbit_heavy
imodbits_bow = imodbit_cracked|imodbit_bent|imodbit_strong|imodbit_masterwork
imodbits_crossbow = imodbit_cracked | imodbit_bent|imodbit_masterwork
imodbits_missile = imodbit_bent|imodbit_large_bag
imodbits_thrown = imodbit_bent|imodbit_heavy|imodbit_balanced|imodbit_large_bag
imodbits_thrown_minus_heavy = imodbit_bent|imodbit_balanced|imodbit_large_bag
imodbits_horse_good = imodbit_spirited|imodbit_heavy
imodbits_good = imodbit_sturdy|imodbit_thick|imodbit_hardened|imodbit_reinforced
imodbits_bad = imodbit_rusty|imodbit_chipped|imodbit_tattered|imodbit_ragged|imodbit_cracked|imodbit_bent
```

A short look reveals the bundle which makes up the constant imodbits_sword:

```
imodbits_sword = imodbit_rusty|imodbit_chipped|imodbit_balanced|imodbit_tempered
```

This results in-game into the appearance of *"rusty"*, *"chipped"*, *"balanced"* and *"tempered"* forms of the newly added sword. The items you find in merchants and in loot will draw their modifiers randomly from the to them assigned imodbits. This means vice versa that modifiers which are not listed in an item's modifier bits will not be considered for merchant inventory or battle loot. For more experienced modders, it is interesting to note that modifiers which aren't listed in an item's modifier bit can still be used with the operation `troop_add_item`. For example, longbows usually come only in *"plain"*, *"bent"* and *"cracked"* forms; but if we were to add a longbow to the player's inventory with the modifier *"balanced"* on it, the player will receive a balanced longbow.

Take note that imods are just a hard-coded way to give boni or penalties to your items since, the effects are hard-coded. While you cannot alter effects you can rename the item modifiers and edit the price factor and the rarity of them. For this, copy the file item_modifiers.txt (path: "game folder/Data/item_modifiers.txt") to the Data folder of your mod (path: "game folder/module/my-mod/Data").[205] It contains lines as follows:

```
imod_plain Plain_%s 1.000000 1.000000
imod_cracked Cracked_%s 0.500000 1.000000
imod_rusty Rusty_%s 0.550000 1.000000
...
imod_large_bag Large_Bag_of_%s 1.900000 0.300000
```

The first field is the imod ID, the second is name, the third is the price modifier with the plain imod as reference (so how many times more expensive than a plain item is an item with the respective modifier) and the fourth field is the rarity. Modify the fields two to four to your likings. While you cannot change the effects of modifiers, you can always add custom effects with triggers on a mission template which give an agent other boni or penalties based on the imod of their items or horse. [206]

There are different opinions about the usage of hard-coded modifiers. Some see no use for them except for the above mentioned built-in loot and merchandise randomization. Others however use them either to create item variants, as shall be explained in the upfollowing paragraphs, or to “factionalize“ items without creating a bunch of variants, especially if one repurposes some unused item modifiers that have no effect on item stats (to which we come later). It might also prove useful where you have just updated graphical resources and the new ones can be marked as the plain default mesh while the outdated ones can be marked by imodbits_bad. The downside is that you can’t use them as equipment overrides or re-equip agents with them or display them on item mesh overlays.[207]

Using Modifiers for Item Variants[208]

Beside their effects on stats and costs, modifiers enable you to have a variety of different skins for a item by adding multiple meshes to the respective tuple fields, referencing a modifier bit out of the ones at tuple field 8. Add it as follows at tuple field 3:

```
3 ["item", "Item", [("mesh",0),("mesh_a",imodbit_rusty),("mesh_b",imodbit_chipped)
      ,("mesh_c",imodbit_balanced)], ... , imodbits_sword],
```

[205] Alternatively you might have a file named module_item_modifiers.py in your Module System if you use a custom one and can compile it into the Data folder. There is also a way to modify the compilation of module_troops.py so that you can assign items with modifiers already to a troop entry. It involves the 0 digit following the item ID, and you need to bitshift the appropriate imod, which is done by some overhauled Module Systems.

[206] Compilation of posts of Somebody, Modding Q&A, Lav, Modding Q&A, Ikaguia, Modding Q&A, kalarhan, Modding Q&A, Modding Q&A and Modding Q&A, and dunde, Modding Q&A, Modding Q&A and Modding Q&A.

[207] See the comments of MadVader and Somebody, Modding Q&A.

[208] Taken partially from maw, Python Script/Scheme Exchange.

Now every time the item is pulled randomly and is imodbit_rusty, it shows up with the mesh_a mesh, every time it randomly appears in a merchants inventory as “balanced“ it is the mesh_c mesh. You get the idea. The modifier `plain` is, as mentioned before, the normal default item, the zero behind the mesh name has actually the same effect as an `imodbit_plain`. You can also assign directly a specific item modifier via an operation. At this place it is recommended to look into (a documented version) of header_operations.py and look up all available operations at which the imodbit can play a role via the keyword `item_modifier`. Take note that you can also add a mesh variant at scabbards (and probably at inventory and flying projectiles too), given the respective item has such:

```
["item", "Item", [("mesh",0),("mesh_scabbard", ixmesh_carry),("mesh_a",
    imodbit_rusty),("mesh_a_scabbard", ixmesh_carry|imodbit_rusty), ...], ... ,
    imodbits_sword],
```

Why does this all come in handy? Well, you can edit an armour or a weapon mesh just slightly to reflect its condition, then associate it this way so it really looks like tagged with that condition. For example you can make a mesh reflecting a ragged leather that actually looks ragged or a rusty sword with rust on it. This mixes up a bit the visuals in-game. By adding all the meshes to one item, they (and their values) are all associated with one item entry. So various swords with roughly the same stats can all be called in the class of 'longsword' or 'shortsword' or 'bastard sword' but allow different visuals.[209]

At Vanilla M&B, so the predecessor of M&B Warband, this functionality could also be used to circumvent the maximum number of items which modders could add to the game. This limit is at 915[210] but is not directly given anymore at Warband. Instead the maximum number of items you can have equals the number of scene props you have defined.[211] Otherwise you will face an indirect item limit in the multiplayer at dropping a weapon: Items dropped on the ground can then change identities but that apparently only happened after the benchmark of 2048 items is hit.[212] It is a known bug, but TaleWorlds refused to fix it because it would break client-server compatibility. Adding dummy scene props will however raise the limit for droppable items.[213]

Armour Modifiers

Cloth and leather can typically receive Tattered, Ragged, Sturdy, Thick or Hardened while metallic armor can be Rusty, Battered, Crude, Thick, Reinforced and Lordly. Cracked is only used on heavy metal armor made from plates.

[209] Keep in mind that you can also make use of inactive or unused modifiers. You can also rename or hide the imodbit prefix in the ui.csv file.

[210] The limit to somewhere between 911 and 919 items, with most reports stating 915. Exceeding it causes the game to crash to the desktop (CTD).

[211] Lumos, Modding Q&A.

[212] Discovered by Watakushi, Modding Q&A, also referenced by Lumos, Modding Q&A.

[213] cmpxchg8b, Modding Q&A.

Modifier	Bonus	Cost
Lordly	+6	+1050 %
Reinforced	+4	+550 %
Hardened	+3	+290 %
Thick	+2	+160 %
Sturdy	+1	+70 %
Crude	-1	-17 %
Battered	-2	-25 %
Ragged	-2	-30 %
Rusty	-3	-45 %
Tattered	-3	-50 %
Cracked	-4	-50 %

Weapon Modifiers

Weapons which have no requirement do not gain one from the Heavy, Strong, or Masterwork modifiers. For instance, while a Masterwork Short Bow has a Power Draw requirement of 5 rather than 1, a Masterwork Hunting Bow, like a standard Hunting Bow, has no Power Draw requirement.

Modifier	Requirement	Speed	Damage	Cost
Masterwork	+4	+1	+5	+1650 %
Tempered	-	-	+4	+670 %
Strong	+2	-3	+3	+360 %
Balanced	-	+3	+3	+250 %
Heavy	+1	-2	+2	+90 %
Chipped	-	-	-1	-28 %
Rusty	-	-	-3	-45 %
Bent	-	-3	-3	-35 %
Cracked	-	-	-5	-50 %

Ammunition Modifiers

The quantity increase is rounded down to the nearest integer, with a minimum of +1. For example, a Large Bag of Arrows will have 34 in a quiver instead of the standard 30: 30*1.15=34.5, the .5 is truncated to leave 34. Thrown Axes normally have four in a bundle, but a Large Bag of Thrown Axes will have five: 4*1.15=4.6; since .6 is less than 1, it receives the minimum value of +1.

Modifier	Quantity	Damage	Cost
Large Bag	+15	-	+90 %
Bent	-	-3	-35 %

Due to their status as ranged and melee weapons as well as ammunition, some throwing weapons can benefit from the Large Bag modifier as well as the Balanced or Heavy modifier. When the Bent modifier is applied, it is taken from the Weapon Modifiers table above, rather than the Ammunition Modifiers table.

Shield Modifiers

These are the only possible shield modifiers. Shields described as "Old", "Heavy", or "Plain" are not modified, but rather a completely different item that can still carry any one modifier listed in the table above.

Contrary to armor, weapon, and projectile, shield modifiers can be changed. If a shield is destroyed in battle, it may become "deformed", losing a positive trait or gaining a negative one.

Modifier	**Durability**	**Resistance**	**Cost**
Reinforced	+83	+4	+550 %
Thick	+47	+2	+160 %
Battered	-26	-2	-25 %
Cracked	-56	-4	-50 %

Horse Modifiers

A horse "killed" in battle may lose its current modifier and become Lame. After the battle, an unused Lame horse in the inventory will recover over time to become a standard one, without any modifier.[214] This also means buying a Lame horse may be a good idea if it stays in the inventory for enough time, resulting in a standard horse later for less than half the price. Purposely killing a Swaybacked horse in order to make it Lame then having it heal back to normal is also a discount compared to regular prices, although there is a risk of losing the horse altogether.

Beneficial modifiers are a costly addition to the more expensive horses and one should be careful when using one in combat as a Lame horse will never recover back into a Champion. There is also the risk of the horse simply dying outright, thus wasting all the money put into it.

Modifier	**Armour**	**Speed**	**Maneuver**	**Charge**	**Hit Points**	**Requirement**	**Cost**
Champion	-	+4	+2	+2	-	+2	+1350 %
Spirited	-	+2	+1	+1	-	-	+550 %
Heavy	+3	-	-	+4	+10	-	+90 %
Stubborn	-	-	-	-	+5	+1	-10 %
Swaybacked	-	-4	-2	-	-	-	-40 %
Lame	-	-10	-5	-	-	-	-60 %
Meek	-	0	-	-	-	-	? %
Timid	-	0	-	-	-	-	? %

Timid and Meek are at Native inactive modifiers for horses and can be activated via their entries in module.ini:

[214] The horse recovery is hard-coded and dictated by the player's Wound Treatment skill. Lumos, Modding Q&A.

```
timid_modifier_speed_bonus = 0
meek_modifier_speed_bonus = 0
```

Changing the values to any other value gives horses with these modifiers the respective speed bonus or penalty.[215]

Food Modifiers

Food modifiers can be applied manually but have also their own effects while the days pass. Following modifiers are given:

```
imod_fresh = 37
imod_day_old = 38
imod_two_day_old = 39
imod_smelling = 40
imod_rotten = 41
```

[215] Effects on the costs have not been tested yet.

The game checks now for some specific food items what their current modifier is and edits it with days passing. For this a trigger is used which can be found in module_simple_triggers.py:

```
# Setting item modifiers for food
(24,
[
  (troop_get_inventory_capacity, ":inv_size", "trp_player"),
  (try_for_range, ":i_slot", 0, ":inv_size"),
      (troop_get_inventory_slot, ":item_id", "trp_player", ":i_slot"),
      (this_or_next|eq, ":item_id", "itm_cattle_meat"),
      (this_or_next|eq, ":item_id", "itm_chicken"),
                   (eq, ":item_id", "itm_pork"),
      (troop_get_inventory_slot_modifier, ":modifier", "trp_player", ":i_slot"),
      (try_begin),
          (ge, ":modifier", imod_fresh),
          (lt, ":modifier", imod_rotten),
          (val_add, ":modifier", 1),
          (troop_set_inventory_slot_modifier, "trp_player", ":i_slot", ":modifier"),
      (else_try),
          (lt, ":modifier", imod_fresh),
          (troop_set_inventory_slot_modifier, "trp_player", ":i_slot", imod_fresh),
      (try_end),
  (try_end),
]),
```

This trigger is getting active every 24 hours. It affects only specific food items (cattle meat, chicken, port) and checks if the item has an modifier with a value between 37 (including the modifier fresh) and 41 (excluding the modifier rotten). If so, the modifier will go on one step, so for example a one-day-old (38) chicken will become a two-day-old (39) chicken. If the food item has no modifier yet at all, the modifier fresh will be set (else_try part) and the item will find its way into the other circulation at the next day.

Unused modifiers

Some extra modifiers exist in the code but are not used. These include: Cheap, Poor, Deadly, Powerful, Exquisite, Rough, Fine, Sharp, Meek, Superb, Old, Timid, Plain, Well-Made. It seems likely that Old and Plain were intended for shields before those words were simply incorporated into specific types of shields. Timid and Meek are for horses and can be activated as mentioned above.

Except imod_fine (adds +1 dmg) they are all deprecated and have no effect on weapons. Since it is not possible to create new weapon modifiers it is however an option to give those unused ones a “second life“ manually via:

```
(troop_get_inventory_slot_modifier, ":item_modifier", "trp_player", ":cur_slot"),
(eq, ":item_modifier", imod_deadly),
```

This way it would be possible to create for example an NPC whose weapon is having a “deadly“ modifier. In a mission templates you can create a ti_on_agent_hit trigger that instantly kills the target if it has less than 50 % hp and if it was hit with mentioned weapon. Of course the “deadly“ modifier can’t appear on weapons on the usual way, so this feature will be kinda unique. It’s just an example but you get the idea.[216]

5.3.4.8 Usage of ixmesh

In header_items.py you will find three ixmesh flags which can be used to assign extra mesh IDs to the item entry:

ixmesh_carry is getting used to assign an extra mesh to the weapon item entry which is getting used when the player hasn’t drawn the weapon and is just carrying it at his body. Usually used for scabbards, bows, throwing weapons, quivers and bolt bags. Take note that scabbards, quivers and bolt bags are making use of vertex animations to work properly, and that ixmesh_carry cannot be used at crossbows.[217]

ixmesh_flying_ammo is typically getting used at arrows, bolts, throwing and if given bullet item entries. With this entry you can control which mesh is getting used for flying projectiles. Usually the flying missile mesh is the same as the weapon, arrow or bolt or is a simpler, less performance heavy shape. Be aware that you are not able to assign particle systems to flying projectiles.[218]

ixmesh_inventory is getting used to set a different mesh for the inventory picture. That way you can use multimeshed or rigged items but still have them oriented and textured correctly in the inventory.[219] The ixmesh_inventory mesh also always matches the icon of the game UI ammo icon which is otherwise showing the main mesh.[220] Additionally it is the mesh which will appear as the spawned item.[221]

[216] Dalion, Mount & Blade Modding Discord.

[217] HokieBT reported such for Vanilla M&B, ixmesh_carry flag doesn’t work for some items?, confirmed at current engine version by Dalion.

[218] MadocComadrin, Modding Q&A.

[219] Mandible, Modding Q&A.

[220] Lumos, Modding Q&A.

[221] dstn, Mount & Blade Modding Discord.

5.3.4.9 Item Extra Description

You might have an item to which you want to add some kind of description, giving the player some additional informations about the item. When you see the weapon information, you can see there the stats and things like *Unbalanced* or *Can crush trough blocks*. For this you need to edit the script `game_get_item_extra_text`.[222] The item description at the popup text is working in a callback style, meaning that the script gets called every time an item detail frame is displayed. It returns a string using

```
(set_result_string,"@this thingie will be displayed at the bottom ^_^"),
```

and optionally you can also specify the custom color by using, for example, `(set_trigger_result, 0x00ff7f)`.[223] You can also set up a bunch of definitions (in module_constants.py) for the range of your newly added items (for example artifacts), and then put their descriptions strings in the same order as the items (in module_strings.py). Then, by finding the offset, you can easily find the description:[224]

```
        (else_try),
          (is_between, ":item_no", artifacts_begin, artifacts_end),
          (try_begin),
            (eq, ":extra_text_id", 0),
            (store_sub, ":offset", ":item_no", artifacts_begin), #get offset
            (val_add, ":offset", "str_atrifact_1"), #get string
            (set_result_string, ":offset"),
            (set_trigger_result, 0xFFEEDD),
          (else_try),
```

5.3.4.10 Item Triggers

At the optional tuple field 9 of the item entry you can add one or a list of simple triggers which are then associated with the item. As the file header_triggers.py shows, there are four different simple triggers possible to be implemented here:[225]

ti_on_init_item contains the trigger parameter 1, the agent ID of the item wearer (-1 if none), and the trigger parameter 2, the troop ID of the item wearer (-1 if none). This trigger will run if

[222] Somebody, Modding Q&A, and Seek n Destroy, Modding Q&A. Latter one gives a nice example for how to implement a new description for an item. At VC you can also use the script `get_item_modifier_effects` that adds the description of the item modifier like *Lordly*, *Lame*, etc. and replace the text or add an own new one for your specific case; kalarhan, Modding Q&A.

[223] Swyter, Modding Q&A.

[224] Somebody, Modding Q&A.

[225] Base informations taken from cdvader, Documentation on Triggers, bundled with additional information snippets taken from kalarhan, VC_Tweaks_Tool header_items.py.

a item is initialized, for example, in the inventory window or in scenes. This simple trigger does not work at horses, boots and gloves which implies that the operations `cur_item_add_mesh` and `cur_item_set_material` will also not work for them. Most commonly used for heraldic mail, heraldic shield and for the torch item.

ti_on_weapon_attack contains the trigger parameter 1, the agent ID of the item wearer, and the position register 1, the weapon items position of the weapon which performed the attack. This trigger will run if a item is used, for example, if you shoot a crossbow. It works on ranged and melee items and gets fired on attack release (so not on left-click but on left-click release).[226] Does not fire on client side in multiplayer missions.

ti_on_missile_hit contains the trigger parameter 1, the agent ID of the shooter who just fired the missile, trigger parameter 2, the object type it hits, and the position register 1, the position of the missile which just landed somewhere. This trigger will run if a missile lands somewhere, for example, if you shoot your bow and the arrow lands. It works only on projectiles (arrows, bolts, throwing) and not on launchers (bows, crossbows, muskets, pistols)[227]. Not all hits will be registered on client-side in multiplayer missions. The different object type parameters are:

0 world (landscape)

1 hostile agent

2 dynamic prop

3 world object (trees, stones, scene interior basic mesh)

4 mission object (most scene props)

8 friendly agent

9 neutral agent

10 under water

You can use this simple trigger to create particle effects when the missile hits the ground, implementing something like

```
        (particle_system_burst,<par_sys_id>, pos1,[percentage_burst_strength]),
```

into the item trigger.[228] Take note that it is not possible to get the instance of a flying missile.[229] Missiles have IDs but they are reused all the time - as soon as a missile hits, it's not valid

[226] Highlander, Modding Q&A.

[227] Throwing weapon count as both, as a projectile and as a launcher, which enables this simple trigger for them.

[228] The_dragon, Modding Q&A.

[229] cmpxchg8b, Modding Q&A.

anymore. In some cases (ground hit, scene prop hit) a mission object gets created, in some others (agent hit, shield hit, spawned prop hit) a mesh gets created, but in any case the actual missile is deleted.[230]

ti_on_shield_hit contains the trigger parameter 1, the agent ID of the receiver who carries the shield, the trigger parameter 2, the agent ID of the dealer who hit the shield, the trigger parameter 3, the inflicted damage on the shield, the trigger parameter 4, the item ID of the weapon (ranged weapon in case of ranged attack), and the trigger parameter 5, the item ID of the missile (ammo in case of ranged attack). If `set_trigger_result` is used in the code, the operation parameter will override the damage dealt to the shield.

All other simple triggers referencing items have to be called within a mission template.

5.3.4.11 Equipment Slots

There are in total ten equipment slots available. This amount is hard-coded into the game engine, you cannot add new ones. You can however create a workaround with slots. Take note that the food slot cannot be repurposed to something else, in Native it is currently deactivated (see Subchapter 5.3.4.15 for the details). The following equipment slots are given.

```
ek_item_0 = 0
ek_item_1 = 1
ek_item_2 = 2
ek_item_3 = 3
ek_head   = 4
ek_body   = 5
ek_foot   = 6
ek_gloves = 7
ek_horse  = 8
ek_food   = 9
```

5.3.4.12 Maximum Inventory Items

In header_items.py you might have found the line `maximum_inventory_items = 96`. It might give you the impression that changing that number increasing the maximum amounts of items which a person can carry in his inventory. It is however just a contant that helps you to calculate the amount of items to offset by when playing with inventories. The actual limit of 96 is hard-coded.[231]

[230] cmpxchg8b, [WB] Warband Script Enhancer v3.2.0.

[231] Somebody, Modding Q&A.

5.3.4.13 Books

If you think about adding new books it is the best to first look up how the other books are getting handled. As mentioned above there are two different kind of books, readable books and reference books. You will also see them sorted like this in the Native Module System:

```
#This book must be at the beginning of readable books
 ["book_tactics","De Re Militari", [...],
 ["book_persuasion","Rhetorica ad Herennium", [...],
 ["book_leadership","The Life of Alixenus the Great", [...],
 ["book_intelligence","Essays on Logic", [...],
 ["book_trade","A Treatise on the Value of Things", [...],
 ["book_weapon_mastery", "On the Art of Fighting with Swords", [...],
 ["book_engineering","Method of Mechanical Theorems", [...],

#Reference books
#This book must be at the beginning of reference books
 ["book_wound_treatment_reference","The Book of Healing", [...],
 ["book_training_reference","Manual of Arms", [...],
 ["book_surgery_reference","The Great Book of Surgery", [...],
```

The two books at the beginning of each book kind have to be there since they are declared as the first item within their respective category in module_constants.py:

```
reference_books_begin = "itm_book_wound_treatment_reference"
reference_books_end    = trade_goods_begin
readable_books_begin = "itm_book_tactics"
readable_books_end    = reference_books_begin
books_begin = readable_books_begin
books_end = reference_books_end
```

The best way to add your own books is to add it at the end of the respective category. Then do a full text search for a similar book to locate the scripts and the simple triggers which handle the books and their effects. You should find then the scripts game_get_item_extra_text and game_get_skill_modifier_for_troop or the simple trigger coming after the comment # Read books if player is resting. There you add then your new books with their effects.

Keep in mind that a book doesn't need to be a book: You can also have a special artifact or such which gives a permanent skill bonus while being carried in the inventory, just like a reference book.

5.3.4.14 Morale Effect of Food[232]

Above we already discussed the existing food modifiers. You might now also want to alter the size of the morale effect of the single food items.[233] For this you can first take a look into the script `get_player_party_morale_values` in module_scripts.py. Filtering for the keyword “food“ leaves you with the following code snippet:

```
1       (assign, "$g_player_party_morale_modifier_food", 0),
2       (try_for_range, ":cur_edible", food_begin, food_end),
3         (call_script, "script_cf_player_has_item_without_modifier", ":cur_edible",
              imod_rotten),
4         (item_get_slot, ":food_bonus", ":cur_edible", slot_item_food_bonus),
5
6         (val_mul, ":food_bonus", 3),
7         (val_div, ":food_bonus", 2),
8
9         (val_add, "$g_player_party_morale_modifier_food", ":food_bonus"),
10      (try_end),
11      (val_add, ":new_morale", "$g_player_party_morale_modifier_food"),
12
13      (try_begin),
14        (eq, "$g_player_party_morale_modifier_food", 0),
15        (assign, "$g_player_party_morale_modifier_no_food", 30),
16        (val_sub, ":new_morale", "$g_player_party_morale_modifier_no_food"),
17      (else_try),
18        (assign, "$g_player_party_morale_modifier_no_food", 0),
19      (try_end),
```

You start out with a food morale modifier of 0 (line 1), so the calculation starts anew each time this script is getting called. For every item within the constants `food_begin` and `food_end -1` (line 2) a script is called to check if the player has such item with a rotten modifier (line 3). If yes, the script fails (cf_scripts are can_fail_scripts), the loop is thus broken for this item and it continues with the next one. If not, the food bonus variable is fetched from `slot_item_food_bonus` (line 4), gets multiplied with 3 and divided by 2 (lines 6 and 7) and then added to the global variable summarizing the food morale modifier (line 9). If the script has looped through all food items it adds the total food morale modifier to the new morale (line 11).

Afterwards a check is getting done: If the player has no suitable food and thus a food morale multiplier of zero (line 14) the player gets a morale penalty of 30 assigned (line 15) which gets

[232] Integrating informations given in posts by Swyter, Modding Q&A, and Somebody, Modding Q&A.

[233] For the consumption rate of food look for the comment `Consuming food at every 14 hours` in module_simple_triggers.py.

substracted of the new morale (line 16). Otherwise the penalty is set to 0 (line 18). If the player has no food in his inventory for multiple days in a row, his morale penalty gets therefore stacked up.

How to edit the morale effect of food? Multiple ways are pointed out above, depending on what you want to achieve. The two most obvious ones are either to edit the declared range of constants or to include new food items within it. To alter the amount of items getting checked you can edit the range of the food items by redefining the constants `food_begin` and/or `food_end` in module_constants.py. To re-enable drinks for example, you can change `food_begin` from `"itm_smoked_fish"`to `"itm_wine"`. Make sure that you have your food items sorted accordingly in module_items.py.

Alternatively you can also alter the individual item food bonus. For this you have to watch out for the slot `slot_item_food_bonus`. That one gets initialized via the script `initialize_item_info`. There you find at the top the for us relevant code lines:

```
# Setting food bonuses - these have been changed to incentivize using historical rations. Bread is the most cost-efficient
#Staples
(item_set_slot, "itm_bread", slot_item_food_bonus, 8), #brought up from 4
(item_set_slot, "itm_grain", slot_item_food_bonus, 2), #new - can be boiled as porridge

#Fat sources - preserved
(item_set_slot, "itm_smoked_fish", slot_item_food_bonus, 4),
(item_set_slot, "itm_dried_meat", slot_item_food_bonus, 5),
(item_set_slot, "itm_cheese", slot_item_food_bonus, 5),
(item_set_slot, "itm_sausages", slot_item_food_bonus, 5),
(item_set_slot, "itm_butter", slot_item_food_bonus, 4), #brought down from 8

#Fat sources - perishable
(item_set_slot, "itm_chicken", slot_item_food_bonus, 8), #brought up from 7
(item_set_slot, "itm_cattle_meat", slot_item_food_bonus, 7), #brought down from 7
(item_set_slot, "itm_pork", slot_item_food_bonus, 6), #brought down from 6

#Produce
(item_set_slot, "itm_raw_olives", slot_item_food_bonus, 1),
(item_set_slot, "itm_cabbages", slot_item_food_bonus, 2),
(item_set_slot, "itm_raw_grapes", slot_item_food_bonus, 3),
(item_set_slot, "itm_apples", slot_item_food_bonus, 4), #brought down from 5

#Sweet items
(item_set_slot, "itm_raw_date_fruit", slot_item_food_bonus, 4), #brought down from 8
(item_set_slot, "itm_honey", slot_item_food_bonus, 6), #brought down from 12

(item_set_slot, "itm_wine", slot_item_food_bonus, 5),
(item_set_slot, "itm_ale", slot_item_food_bonus, 4),
```

You can also see at the comments that the developer tried to bring some systematic into it and documented some little changes here. You can now either edit those numbers to be more to your liking and/or add respective lines of newly added food items and/or delete them vice versa.

5.3.4.15 Food slot

foot slot spits out certain item (recheck since Somebody mentions out of order), check also if only specific things can be thrown into the food slot or everything. Hard-coded items[234]

5.3.4.16 Other Notes

Need to integrate them to the respective categories.

Riders with daggers don't fight, daggers can't block[235] What exactly defines a dagger, itc flag or weapon length?

arcs and heights throwing items[236], range of weapon[237]

Projectile distance damage power with some testings at upfollowing page[238]

damage value determines accuracy[239], more accurate distance weapon[240]

max weight of items[241]

attack speed[242]

crush through[243], can crush through blocks[244]

no left handed weapons.[245]

equipment and inventory slots[246]

spears longer than 2,4 m need custom hitbox?[247]

hard-coded horse crippling[248]

Need to write down for the guide how power draw requirement limits the use of power draw skill. It's basically written at the list of available module skills but it should be marked more clear at the item section that a higher power draw requirement has a direct effect on this.

consumable items[249]

bonus items requirements[250]

[234] Somebody, Modding Q&A.
[235] Somebody and dunde, Modding Q&A.
[236] DanyEle, Modding Q&A.
[237] kraggrim (credit), Modding Q&A.
[238] Discussion and test, Modding Q&A.
[239] Somebody, Modding Q&A.
[240] Zirkhovsky and MadocComadrin, Modding Q&A, and Willhelm (credit), Modding Q&A.
[241] Vornne, Modding Q&A.
[242] Lav, Modding Q&A.
[243] Ikaguia (credit), Modding Q&A.
[244] Somebody, Modding Q&A.
[245] Somebody, Modding Q&A.
[246] kalarhan, Modding Q&A.
[247] jacobhinds, Modding Q&A.
[248] jacobhinds, Modding Q&A.
[249] kalarhan, Modding Q&A.
[250] kalarhan, Modding Q&A and Modding Q&A.

item Abundance value[251]

weapon unsheathable[252]

5.3.4.17 Items as Scene Props

Note: Place here code lines of Sebastian which do automatically increase the number of sceneprops.

[251] Tegan (credit), How do I add items to merchants?.

[252] Somebody, Modding Q&A.

5.3.5 Module Animations

```
There are two animation arrays (one for human and one for horse). Each animation in these arrays contains the following fields:
1) Animation id (string): used for referencing animations in other files. The prefix anim_ is automatically added before each animation-id
2) Animation flags: could be anything beginning with acf_ defined in header_animations.py
3) Animation master flags: could be anything beginning with amf_ defined in header_animations.py
4) Animation sequences (list).
4.1) Duration of the sequence. Basically the speed of the animation, the higher the number, the slower the animation will progress.
4.2) Name of the animation resource.
4.3) Beginning frame of the sequence within the animation resource.
4.4) Ending frame of the sequence within the animation resource.
4.5) Sequence flags: could be anything beginning with arf_ defined in header_animations.py
4.6) Sound Timing (float): Optional. Used to play sounds in animations with arf_make_walk_sound and arf_make_custom_sound.
     Can be used in conjunction with pack2f or pack4f to make the animation
     play the sound multiple times. Timing assumed to be percentages of
     animation completion. For example a timing of 0.0 will play at the start
     while 1.0 will play at the end.
4.7) Offset Position (float, float, float): Optional. Used to move the animating agent's position at the end of the animation.
                                                   Requires acf_displace_position flag
4.8) Unknown Float. Optional. Speculated to be anim delay, but research is so far inconclusive
```

module_animations.py is where the animation of the module are getting recorded and defined. The file has a declaration of constants before the usual list of tuples begins.

The animations are semi-hard-coded: Many animations are hard-coded such way inside the game engine that their position (if its #1, #100, etc.) matters. However, while the game engine expects animation #30 to do something when a particular action is used (like a block) you can still edit the individual sequences, resources or times and thus define your own visual animation and special flags. Important is that each animation must stay at the same position, otherwise the game won't run properly! You must not change the IDs of used animations![253]

[253] Mixed description of Native MS with some words of kalarhan, Modding Q&A.

To add an animation, you need to overwrite one of the “unused“ animations.[254] Those are place holder animations and thus free, and can be used for stuff like a special animation for sitting, for a npc working on a field, etc. Stuff you will define by hand and for which you have a suitable animation resource at hand. Also keep in mind there are animation for humans and animations for horses, don’t mix them. If the animation is for humans, you need to replace one of the `anim_human_unused`-lines and if it is for horses, replace one of the `anim_horse_unused`-lines. So comment out the first available “unused“ animation and add or paste yours above or below it. You can also change the ID name of your animation to your likings.[255]

At adding new animations you need to provide a set of flags to control the execution, frames, speed, sync, if it should be on a cycle or not, the effect on rigging and how to blend in with the next animation.[256] Unfortunately the official documentation lacks details, so we try to slowly fill out the gaps of knowledge about the meaning of some of the flags.

5.3.5.1 Animation Capability Flags

The following animation capability flags are available for usage:[257]

acf_align_with_ground aligns the agent with the angle of the ground.

acf_displace_position is getting used in conjunction with tuple 4.7 to offset the agent at the end of the animation.[258]

acf_enforce_all Up4research.

acf_enforce_lowerbody Up4research.

acf_enforce_rightside rotates the right arm to the look direction (more than the rest of the upper body).

acf_ignore_slope forces the agent (obiously horses) to not align with the ground? Up4research.

acf_left_cut Up4research.

acf_lock_camera gets used for falling animations, Up4research.

[254] This is actually not true since you can actually add animations at the bottom (sorting at the existing ones is still important though!) and they will work just fine. The possible amount which can get added is unlimited for the Singleplayer and about 800 max for the Multiplayer. cmpxchg8b, Warband Script Enhancer v3.2.0. For beginners it is however advised to start with overwriting the unused animation entries.

[255] Remarks of Caba‘drin, Modding Q&A, and kalarhan, Modding Q&A.

[256] kalarhan, Modding Q&A.

[257] Working in many remarks of _Sebastion_, Animation flags and their usage.

[258] cmpxchg8b, Undocumented fields in module_animations.py.

acf_overswing Up4research.

acf_parallels_for_look_slope seems to rotate the upper body towards the look direction as well. Up4research.

acf_right_cut Up4research.

acf_rot_vertical_bow Up4research.

acf_rot_vertical_sword Up4research.

acf_rotate_body a Vanilla M&B flag, it might work in the same way as `acf_displace_position` but for rotation? Up4research.

acf_synch_with_horse synchronises the rider animation with that of their mount.

acf_thrust Up4research.

5.3.5.2 Animation Master Flags

The following animation master flags are available for usage:

amf_accurate_body up4research.

amf_client_owner_prediction sends more precise animation info, like direction and frame to the server, for accurately combat calculations? Up4research.

amf_client_prediction forces the animation to not sync with the server, or vise versa. Up4research.

amf_continue_to_next plays sequencially the upfollowing animation when the current one has ended, regardless of its priority.

amf_hide_weapon unused in Native and DLCs, up4research.

amf_keep lets the animation hold at the last frame of the animation until an animation of equal or greater priority gets called.[259]

amf_play lets the animation play through and end when reaching the final frame.

amf_restart restarts the animation even if it is the current animation.

amf_rider_rot_bow is probably limiting the rotation of a mounted agent to some unknown degree.

[259] Using it together with arf_cyclic while having a variety of animation resources will probably let the agent stick to a specific randomly chosen animation until the agent switches to a different action. Up4research.

amf_rider_rot_couched_lance is probably limiting the rotation of a mounted agent to some unknown degree.

amf_rider_rot_crossbow is probably limiting the rotation of a mounted agent to some unknown degree.

amf_rider_rot_defend is probably limiting the rotation of a mounted agent to some unknown degree.

amf_rider_rot_overswing is probably limiting the rotation of a mounted agent to some unknown degree.

amf_rider_rot_pistol is probably limiting the rotation of a mounted agent to some unknown degree.

amf_rider_rot_shield is probably limiting the rotation of a mounted agent to some unknown degree.

amf_rider_rot_swing_left is probably limiting the rotation of a mounted agent to some unknown degree.

amf_rider_rot_swing_right is probably limiting the rotation of a mounted agent to some unknown degree.

amf_rider_rot_throw is probably limiting the rotation of a mounted agent to some unknown degree.

amf_rider_rot_thrust is probably limiting the rotation of a mounted agent to some unknown degree.

amf_start_instantly prevents that the animation needs to wait to start and interrupts all current animations to play.

amf_use_cycle_period repeats the same animation if multiple animation resources are tied to a single action? Up4research.

amf_use_defend_speed lets the defend speed of the used item influence the time it takes for the animation? Up4research.

amf_use_inertia slows down the animation related to weapon mass? Up4research.

amf_use_weapon_speed lets the weapon speed of the used item influence the time it takes for the animation? Up4research.

Restrict rotation when mounted[260]

amf_client_prediction doesn't let animation sync.[261]

amf_use_weapon_speed[262]

horse animation[263]

Integrate custom declared animation priority flags with notes of dstn.

give animation higher priority.[264]

animation higher priority[265]

custom mount animation and also some note about priority[266]

5.3.5.3 Animation Sequence Flags

The following animation sequence flags are available for usage:

arf_blend_in_x lets blend in an animation to the one currently playing. It is currently unknown if this is in frames, or some unknown measurement of time, up4research.

arf_cyclic causes the animation sequence to loop.

arf_lancer up4research.

arf_make_custom_sound lets a sound be played when specific animations are getting triggered. The animations as well as the connected sounds are hardcoded, modders can basically just toggle the sound on or off by setting or removing this flag. The animation triggers and sound connections are as follows:

This flag is getting used together with the pack2f function in tuple 4.6. The first value of it determines at which percentage of the animation completion the first occasion sound (Get new ammo) is getting played, the second value the trigger moment of the second occasion sound (Reload ranged weapon) is getting played. The third occasion (Agent falling) has only one sound stage, so the second value is always zero.

arf_make_walk_sound up4research.

[260] cmpxchg8b, Modding Q&A.

[261] cmpxchg8b, Modding Q&A.

[262] Shredzorz (credit), [WB] Warband Script Enhancer v3.2.0.

[263] Caba‘drin, Modding Q&A.

[264] Somebody, Modding Q&A.

[265] dunde, Modding Q&A.

[266] The_dragon, Modding Q&A and Modding Q&A.

Animation	Sound
1) Get new ammo	
anim_ready_bow	snd_pull_arrow
anim_reload_crossbow	snd_reload_crossbow
anim_reload_crossbow_horseback	snd_reload_crossbow
anim_reload_musket	snd_reload_crossbow
anim_reload_musket_full	snd_reload_crossbow
2) Reload ranged weapon	
anim_ready_bow	snd_pull_bow
anim_reload_crossbow	snd_reload_crossbow_continue
anim_reload_crossbow_horseback	snd_reload_crossbow_continue
3) Agent falling	
anim_fall_face_hold	snd_body_fall_small
anim_fall_chest_front	snd_body_fall_small
anim_fall_abdomen_hold_front	snd_body_fall_small
anim_fall_head_front	snd_body_fall_small
anim_fall_right_front	snd_body_fall_small
anim_fall_body_back	snd_body_fall_small
anim_fall_rider_right_forward	snd_body_fall_small
anim_fall_rider_right	snd_body_fall_small
anim_fall_rider_left	snd_body_fall_small
anim_rider_fall_right	snd_body_fall_small
anim_rider_fall_roll	snd_body_fall_small
anim_strike_fall_back_rise	snd_body_fall_small
anim_horse_fall_right	snd_body_fall_small
anim_horse_fall_roll	snd_body_fall_small

arf_use_stand_progress lets the animation use the current stand animation progress? up4research.

arf_stick_item_to_left_hand up4research.

arf_two_handed_blade up4research.

arf_use_walk_progress lets the animation use the current walk animation progress or agent speed affects the animation speed. Up4research.

arf_use_inv_walk_progress same as `arf_use_walk_progress`, but plays the animation in reverse. Up4research.

Stick item to left hand animation problem[267]

5.3.5.4 Upper and Lower Body at Animations

upper and lower body at animation[268]

[267] Somebody, Modding Q&A.

[268] Example, Modding Q&A.

uperbody part animation[269]

5.3.5.5 Variations for Animations

It is possible to add different variations of the same animation, just put them in the list of animation reference entries. If the animation has multiple parts and you have variations, they might however not sync up properly during the transitions.[270]

5.3.5.6 Animation Sounds

The settings pack2f and pack4f (pack two/four floats) are assigning sound to the animations, defining at what percentage of the animation a sound is getting played. If more than two values are needed, pack4f is used instead of pack2f. How many and what sounds are actually played is almost entirely hardcoded.[271] See the descriptions of the animation sequence flags `arf_make_custom_sound` and `arf_make_walk_sound` for more details.

5.3.5.7 Ragdolls

Entry field 4.8 contains the the time during the animation that the ragdoll effects will begin, used at the death animations in Native. You can disable ragdolls in the Options in-game.[272]

5.3.5.8 Skin/Race-specific Animations

Unfortunately, specific movement or combat animations for custom races are not possible without extremely inefficient workarounds, i.e. working out when each animation would play (if the agent is further left than it was in the last frame, use the move left animation). It is made harder by the fact that for some silly reason animations have to be assigned at every frame, which also causes lag.[273]

Fun fact: After "researching" the engine it's quite obvious that the developers planned to have per-agent action sets but for some reason they scrapped the idea without even removing the code.[274]

5.3.5.9 Animations and Item Types

The walking and holding animations are influenced by the item types of the weapons an agent is carrying and, to some lesser degree, by the item property and capability flags of the prementioned

[269] jacobhinds, Modding Q&A.
[270] Somebody, Modding Q&A.
[271] cmpxchg8b, Modding Q&A.
[272] Ruthven, Modding Q&A.
[273] Jacobhinds, Race specific animation.
[274] cmpxchg8b, Suggestions/requests thread – post yours here.

weapons. For this you will find first a description of each of the four cases and then, sorted by the cases, the animations connected to each case.

Case 0 is the default case, basically used for unarmed agents.

Case 1 gets gets applied if an agent carries 1) an item of either the type `itp_type_crossbow` or `itp_type_musket` AND with the item capability flag `itcf_reload_pistol` or 2) every other not in case 2 or 3 mentioned combination. Basically, this case contains all one-handed weapons (and whenever the agent carries a shield too) as well as all kind of throwing weapons, bows and pistols.

Case 2 gets applied if an agent carries 1) an item with the flag `itp_type_polearm` AND no secondary item in the left hand or 2) an item of either the type `itp_type_crossbow` or `itp_type_musket` AND without the capability flag `itcf_reload_pistol`. Basically, this case contains all polearms, crossbows and muskets.

Case 3 gets applied if an agent carries an item of either the type `itp_type_one_handed`, `itp_type_two_handed` or `itp_type_polearm` AND with the item property flag `itp_two_handed`. Basically, this case contains all two-handed weapons.

The connection between item flags and the running and walking animations is a simple one as can be seen in the table below. Take note that the * is replacing the direction of the animation, so backward, forward, left or right (or a combination of them like forward left etc.).[275]

Case	Animation
Walking	
0	anim_walk_*, m_walkActionNo(?)
1	anim_walk_*_onehanded
2	anim_walk_*_staff and anim_walk_*_polearm
3	anim_walk_*_greatsword and anim_walk_*_twohanded
Common	anim_walk_*_hips_left/right
Running	
0	anim_run_*
1	anim_run_*_onehanded
2	anim_run_*_staff and anim_run_*_polearm
3	anim_run_*_greatsword and anim_run_*_twohanded
Common	anim_run_*_hips_left/right
Turning	
0	anim_turn_right/left
1	anim_turn_right/left_single
2	anim_turn_right/left_staff
3	anim_turn_right/left_greatsword

[275] It seems like the developers didn't watch out for their own naming conventions, resulting in mixed uses of polearms/staff or greatsword/twohanded, even within the same animation category.

The standing animations are also depending on the item types and item property flags as well as if the agent is idle or crouching:

Case	Animation and
Case 0	*IF Alarmed Agent OR Player in Battle OR mtf_team_fight OR Multiplayer*
Crouch	anim_crouch_unarmed
Idle	anim_stand_unarmed
Case 1	
Crouch	anim_crouch_single
Idle	anim_stand_single
Case 2	*IF itp_crossbow*
Crouch	anim_crouch_crossbow
Idle	anim_stand_crossbow
	IF itp_is_pike AND turn amount < MAX(brace_rotation_limit; 0.01)
Crouch	anim_crouch_pike
	Else
Crouch	anim_crouch_staff
Idle	anim_stand_staff
Case 3	
Crouch	anim_crouch_greatsword
Idle	anim_stand_greatsword

5.3.5.10 Animations and Item Capability Flags

The engine is hard-coded to allocate only certain types of combat animations to items with certain item capability flags. You can always try combining existing animations to clear up slots or use up the sets that aren't in use in your module, like the pistol and musket animations in Native. Shield bash, horn blowing and other actions aren't weapon animations - they take up one of the unused human animation entries and are called on-demand by a trigger.[276]

The following tables list which animations are connected with which item capability flags, starting with the defending animations. Keep in mind that the animation entries can be named different than the animations inside the brf files. Look up always the name of the animation resource at the tuple entry of the mentioned animation. Take note that the normal defending animations are getting overruled by the shield animations if the agent wears a shield, and are getting overruled by the fist defending animations if no weapons are given at all:

itcf	Defend Action	Keep Action
Shield overrules	anim_defend_shield	anim_defend_shield_keep
Unarmed defense	anim_defend_fist	anim_defend_fist_keep

If the agent has no shield the defending animations are as follows:[277]

itcf	Defend Action	Keep Action
Defend thrust		
itcf_parry_forward_onehanded	defend_forward_onehanded	defend_forward_onehanded_keep
itcf_parry_forward_twohanded	defend_forward_greatsword	defend_forward_greatsword_keep
itcf_parry_forward_polearm	defend_forward_staff	defend_forward_staff_keep
Defend right		
itcf_parry_right_onehanded	defend_right_onehanded	defend_right_onehanded_keep
itcf_parry_right_twohanded	defend_right_twohanded	defend_right_twohanded_keep
itcf_parry_right_polearm	defend_right_staff	defend_right_staff_keep
Defend left		
itcf_parry_left_onehanded	defend_left_onehanded	defend_left_onehanded_keep
itcf_parry_left_twohanded	defend_left_twohanded	defend_left_twohanded_keep
itcf_parry_left_polearm	defend_left_staff	defend_left_staff_keep
Defend up		
itcf_parry_up_onehanded	defend_up_onehanded	defend_up_onehanded_keep
itcf_parry_up_twohanded	defend_up_twohanded	defend_up_twohanded_keep
itcf_parry_up_polearm	defend_up_staff	defend_up_staff_keep

[276] Somebody, Modding Q&A.

[277] The prefix `anim_*` has been removed from the animation names due to shortage of space.

For ranged weapons the itcf determine the ready and release animations as follows:[278]

itcf	Ready Action	Release Action
itcf_shoot_bow	anim_ready_bow	anim_release_bow
itcf_shoot_crossbow	anim_ready_crossbow	anim_release_crossbow
itcf_shoot_pistol if crouching	anim_ready_pistol anim_crouch_ready_pistol	anim_release_pistol anim_crouch_release_pistol
itcf_shoot_musket	anim_ready_musket	anim_release_musket
itcf_throw_stone	anim_ready_stone	anim_release_stone
itcf_throw_knife	anim_ready_throwing_knife	anim_release_throwing_knife
itcf_throw_axe	anim_ready_throwing_axe	anim_release_throwing_axe
itcf_throw_javelin	anim_ready_javelin	anim_release_javelin

Crossbows, muskets and pistols are the only ranged items which have a separate reloading animation, at the others it is integrated into the ready action.[279]

itcf	Reloading Action
itcf_shoot_crossbow if mounted AND <no itp_cant_reload_on_horseback>	anim_reload_crossbow anim_reload_crossbow_horseback
itcf_reload_pistol if <use_phased_reload = 1> and reload interrupted at middle	anim_reload_pistol anim_reload_pistol_half
itcf_reload_musket if <use_phased_reload = 1> and reload interrupted	anim_reload_musket anim_reload_musket_full

[278] The exact effect of `itcf_shoot_javelin` is still up for research. It seems to flip the physical model of the weapon backwards so that one needs to use a separate jav-variant for it.

[279] If `use_phased_reload` is set to 1 in module.ini, the reloading of muskets and guns can get done with intermediate steps.

For close combat weapons the itcf determine the ready and release animations for unmounted agents as follows:[280]

Ready Action	Release Action
Thrust	
itcf_thrust_twohanded AND <no shield>	
anim_ready_thrust_twohanded	anim_release_thrust_twohanded
itcf_thrust_polearm AND <no shield>	
anim_ready_thrust_staff	anim_release_thrust_staff
itcf_thrust_onehanded AND (<shield> OR <no itcf_thrust_polearm>)	
anim_ready_thrust_onehanded	anim_release_thrust_onehanded
itcf_thrust_onehanded_lance AND (<shield> OR <no itcf_thrust_polearm>)	
anim_ready_thrust_onehanded_lance	anim_release_thrust_onehanded_lance
itcf_thrust_musket AND (<shield> OR <no itcf_thrust_polearm>)	
anim_ready_thrust_musket	anim_release_thrust_musket
Overswing	
itcf_overswing_twohanded AND <no shield>	
anim_ready_overswing_twohanded	anim_release_overswing_twohanded
itcf_overswing_polearm AND <no shield>	
anim_ready_overswing_staff	anim_release_overswing_staff
itcf_overswing_onehanded	
anim_ready_overswing_onehanded	anim_release_overswing_onehanded
itcf_overswing_spear AND <no shield>	
anim_ready_overswing_spear	anim_release_overswing_spear
itcf_overswing_musket AND (<shield> OR <no itcf_overswing_polearm>)	
anim_ready_overswing_musket	anim_release_overswing_musket
Swing left	
itcf_slashleft_twohanded AND <no shield>	
anim_ready_slashleft_twohanded	anim_release_slashleft_twohanded
itcf_slashleft_polearm AND <no shield>	
anim_ready_slashleft_staff	anim_release_slashleft_staff
itcf_slashleft_onehanded	
anim_ready_slashleft_onehanded	anim_release_slashleft_onehanded
Swing right	
itcf_slashright_twohanded AND <no shield>	
anim_ready_slashright_twohanded	anim_release_slashright_twohanded
itcf_slashright_polearm AND <no shield>	
anim_ready_slashright_staff	anim_release_slashright_staff
itcf_slashright_onehanded	
anim_ready_slashright_onehanded	anim_release_slashright_onehanded

[280] Pay attention to the different conditions alongside the itcf!

For mounted agents the ready and release animations are as follows:

Ready Action	Release Action
Thrust	
itcf_thrust_onehanded	
anim_ready_thrust_onehanded_horseback	anim_release_thrust_onehanded_horseback
itcf_thrust_onehanded_lance	
anim_ready_thrust_onehanded_lance	anim_release_thrust_onehanded_lance
Overswing	
itcf_overswing_onehanded	
anim_ready_overswing_onehanded	anim_release_overswing_onehanded
Swing left	
itcf_horseback_slashleft_onehanded AND itcf_horseback_slash_polearm	
anim_ready_slash_horseback_polearm_left	anim_release_slash_horseback_polearm_left
itcf_horseback_slashleft_onehanded	
anim_ready_slash_horseback_left	anim_release_slash_horseback_left
Swing right	
itcf_horseback_slashright_onehanded AND itcf_horseback_slash_polearm	
anim_ready_slash_horseback_polearm_right	anim_release_slash_horseback_polearm_right
itcf_horseback_slashright_onehanded	
anim_ready_slash_horseback_right	anim_release_slash_horseback_right

Might be that I missed all the parry animations :P

Changing blocking speed[281]

thrust_onehanded_horseback animation[282]

bow mounted inactive animations[283]

unequip animations would be interesting to know the relation/connection to flags.

5.3.5.11 Other notes

some constants are getting declared at the beginning (a bit strange, need to review en detail)

Creating animation test[284]

set animation in mp[285]

agent set animation for horse[286]

Some jumping animation info[287]

knockdown animation[288]

[281] Somebody, Modding Q&A.

[282] Mammoet and Somebody, Modding Q&A and Modding Q&A.

[283] jacobhinds, Modding Q&A, and rgcotl (credit), Modding Q&A.

[284] xenoargh, Modding Q&A.

[285] Vornne and MadocComadrin, Modding Q&A.

[286] Somebody, Modding Q&A.

[287] Lumos, Modding Q&A.

[288] _Sebastian_, Modding Q&A.

stop cyclic animation[289]

kill player at certain frame of animation[290]

it is possible to use an engine trick in WB to shoot a pistol through a shield, using a custom animation[291]

[289] _Sebastian_, Modding Q&A.

[290] The_dragon, Modding Q&A.

[291] Darth Mongol the Unwise, Mount & Blade Modding Discord.

5.4 Scene-related Module System Files

5.4.1 Module Scene Props

```
Each scene prop record contains the following fields:
1) Scene prop id: used for referencing scene props in other files. The prefix spr_
    is automatically added before each scene prop id.
2) Scene prop flags. See header_scene_props.py for a list of available flags
3) Mesh name: Name of the mesh.
4) Physics object name: Name of the collision object associated with the mesh.
5) Triggers: Simple triggers that are associated with the scene prop.
```

Add some intro text here

5.4.1.1 Scene Prop Flags

sokf_add_fire Unused in Native, WFaS, VC and NW. Eventually deprecated.

sokf_add_light Unused in Native, WFaS, VC and NW. Eventually deprecated.

sokf_add_smoke Unused in Native, WFaS, VC and NW. Eventually deprecated.

sokf_destructible makes the scene prop destructible/destroyable.

sokf_dont_move_agent_over Speculation: Moving scene props are not affecting the agents?

sokf_dynamic up4research.

sokf_dynamic_physics up4research.

sokf_enforce_shadows up4research.

sokf_face_player lets the scene prop always face the player.[292]

sokf_handle_as_flora up4research.

sokf_invisible makes the scene prop invisible. Typically used for invisible barriers and light scene props.

sokf_missiles_not_attached prevents missiles from sticking in the scene prop.[293]

[292] cmpxchg8b, Modding Q&A.

[293] Somebody, Modding Q&A.

sokf_moveable enables movements for the scene prop - together with its collision mesh[294] - which also get automatically synced with multiplayer clients.[295] The flag prevents the scene prop from casting shadows.[296]

sokf_place_at_origin positiones an already placed scene prop such way to always be at entry point 0.[297]. Typically used for the player's inventory.

sokf_show_hit_point_bar shows the remaining hit points of the destructible scene prop.

sokf_static_movement has been added for NW. The server runs code in the game engine to detect whether players are looking at a usable prop which gets impacted when the prop is also moveable. The flag reduces the performance impact of this combination if the scene prop movement and collision detection is not very important (like a door that just moves between a few positions, as opposed to a ship with agents on it who wouldn't want to fall off).[298]

sokf_type_ai_limiter marks scene props which prevent movement of the AI but not of the player.

sokf_type_ai_limiter3d same but for 3D objects? Used in VC, up4research.

sokf_type_barrier marks a simple 2d barrier, typically an invisible sceneprop, to restrict the player movements at the scene(?).

sokf_type_barrier_leave marks a simple 2d barrier, typically an invisible sceneprop, to restrict the player movements at the scene and at which the scene can be leaved via the town menu(?).

sokf_type_barrier3d marks a simple 3d barrier, typically an invisible sceneprop, to restrict the player movements at the scene(?).

sokf_type_container determines that the scene prop can be used as a chest.

sokf_type_ladder causes slow movements on the scene prop.

sokf_type_player_limiter up4research.

spanim_linear up4research.

spanim_loop_linear up4research.

spr_hit_points(x) determines how many hitpoints that destructible scene prop has.

[294] cmpxchg8b, Modding Q&A.

[295] Somebody, Modding Q&A.

[296] _Sebastian_, Modding Q&A.

[297] Somebody, Modding Q&A.

[298] Vornne, Map Stability and Crashes.

spr_use_time(x) determines how long it takes to use the scene prop. Typically used in multiplayer, make sure to set `can_use_scene_props_in_single_player = 1` in module.ini to make scene props useable in singleplayer too. Keep in mind that a scene prop needs to have a collision mesh for this to work.[299] This flag is also need to be set to make use of the trigger `ti_on_scene_prop_use`.[300]

5.4.1.2 Destroyable Scene Props

Destroyable scene props.[301]

5.4.1.3 Using Scene Props

getting using information text[302]

disable scene prop use[303]

5.4.1.4 Moving Scene Props

Moving sceneprops in MP[304]

5.4.1.5 Vertex-animated Scene Props

vertex animating scene prop[305]

vertex animated scene prop[306]

Usage of prop_instance_deform_in_range for operating the vertex animation seen in brf[307]

5.4.1.6 Light Source Scene Props

Keep in mind that there is a limit of 10 light sources for exterior and 4 for interior scenes. You can place many more but they will not paint the terrain. The player model will still reflect light though.[308] Work in this remark and this remark, as well as this potential workaround.

max point lights[309]

[299] Sinisterius, Modding Q&A.
[300] _Sebastian_, Modding Q&A.
[301] ithilienranger, Modding Q&A.
[302] dunde, Modding Q&A.
[303] Somebody, Modding Q&A.
[304] Vornne, Modding Q&A.
[305] Lumos, Modding Q&A, meanwhile it is possible though, Dalion.
[306] HyperCharge (credit), Animating - import / export.
[307] Dalion, Mount & Blade Modding Discord.
[308] The Bowman, Mount & Blade Modding Discord.
[309] cmpxchg8b, [WB] Warband Script Enhancer v3.2.0 and [WB] Warband Script Enhancer v3.2.0.

maximum light stuff[310]

Scene maximum lightsources[311]

5.4.1.7 Shadows on Scene Props

shadow engine bug (module scene props, connected with collision mesh iirc)[312]

agent shadow[313]

5.4.1.8 Water River Scene Props

water river props[314]

5.4.1.9 Horses and Stairs

Let horses walk stairs[315]

5.4.1.10 Scene Prop Variables

Maximum value for each of both variables is 127.[316] A more general explanation about use of them is also needed.

5.4.1.11 Scene Prop Variants - Alternative Materials

Alternative material for scene props[317]

example code, scene prop prefab[318]

5.4.1.12 Scene Prop Meta Types, Sub Kind ID and Variation IDs[319]

Internally, most items on the scene are instances of the same class (mission object) and are even stored in the same collection. What distinguishes them is meta type, sub kind id and two variation ids. Meta type identifies the type of the object. You can distinguish between the following:

Meta Type 0: Scene props (placed in Editor)

[310] Somebody and cmpxchg8b, Modding Q&A.

[311] Ruthven and _Sebastian_, Modding Q&A, and _Sebastian_, Modding Q&A.

[312] _Sebastian_, Modding Q&A.

[313] cmpxchg8b, Modding Q&A.

[314] _Sebastian_ and InVain, Modding Q&A, Gotha (credit), problem with river_water, Marko (credit), Modding Q&A, and InVain and Marko, Modding Q&A.

[315] NPC99, Modding Q&A.

[316] Dalion, Mount & Blade Modding Discord.

[317] Lav, Modding Q&A, Antonis and La Grandmaster, Modding Q&A and Modding Q&A.

[318] dstn, Mount & Blade Modding Discord.

[319] Bundeling of informations given by cmpxchg8b, Modding Q&A.

Meta Type 1: Entry points

Meta Type 2: Scene items (placed in Editor)

Meta Type 3: No idea what this is

Meta Type 4: Flora (trees, rocks, ...)

Meta Type 5: Passages

Meta Type 6: Spawned items (dropped by player)

Meta Type 7: Spawned items (created on missile hit)

Meta Type 8: Spawned items (dropped on death)

Meta Type 9: No idea what this is

Meta Type 10: Spawned scene props

The sub kind id is the id of the underlying item (item kind id for items, scene prop id for scene props, etc.). The variation ids are used for various stuff, e.g. for items that hold item mesh variation, for passages menu id, etc. You might have seen or will perhaps see a server error like this:

```
Synchronization with server failed: trying to recreate existing object (id: %d,
    meta_type: %d, sub_kind_id: %d, variation_id: %d) with meta_type: %d,
    sub_kind_id: %d, variation_id: %d, do_prune: %d
```

Now you know at least roughly what those fields mean.

Strangely `sokf_missiles_not_attached` works for scene props placed in the Editor (meta type 0) while the developers noted that it only works for dynamic mission objects. Dynamic mission objects are items that you drop (meta type 6), items that you drop on death (meta type 8), items created by missiles when they hit something (meta type 7) and spawned scene props (meta type 10).

5.4.1.13 Other notes

One sided collision mesh[320]

hard-coded scene props?[321]

[320] Somebody, Modding Q&A.

[321] Somebody, Modding Q&A.

5.4.2 Module Scenes

```
 Each scene record contains the following fields:
 1) Scene id {string}: used for referencing scenes in other files. The prefix scn_ is automatically added before each scene-id.
 2) Scene flags {int}. See header_scenes.py for a list of available flags
 3) Mesh name {string}: This is used for indoor scenes only. Use the keyword "none" for outdoor scenes.
 4) Body name {string}: This is used for indoor scenes only. Use the keyword "none" for outdoor scenes.
 5) Min-pos {(float,float)}: minimum (x,y) coordinate. Player can't move beyond this limit.
 6) Max-pos {(float,float)}: maximum (x,y) coordinate. Player can't move beyond this limit.
 7) Water-level {float}.
 8) Terrain code {string}: You can obtain the terrain code by copying it from the terrain generator screen
 9) List of other scenes accessible from this scene {list of strings}.
    (deprecated. This will probably be removed in future versions of the module system)
    (In the new system passages are used to travel between scenes and the passage's variation-no is used to select the game menu item that the passage leads to.)
10) List of chest-troops used in this scene {list of strings}. You can access chests by placing them in edit mode.
    The chest's variation-no is used with this list for selecting which troop's inventory it will access.
```

Pages 23-25

5.4.2.1 Scene Flags

sf_indoors removes the skybox and lighting by sun for this scene.

sf_force_skybox forces adding a skybox even if indoors flag is set.

sf_generate generates terrain by terrain-generator.

sf_randomize randomizes terrain generator key.

sf_auto_entry_points automatically places entry points in randomly generated scenes.[322] Does it place them only at randomly generated scene or also at others?

[322] GetAssista, Modding Q&A, and Docm30, Modding Q&A.

sf_no_horses is a scene marker for the operation `scene_allows_mounted_units`. Does nothing on its own(?).

sf_muddy_water changes the shader of the river mesh.

5.4.2.2 Scene Borders

min/max border scenes[323]

internal and external border[324]

Scene entry numbers deprecated?[325]

5.4.2.3 Water Level

water level modifier (not working?)[326]

water splash sounds and slow speed, question 2[327]

5.4.2.4 Terrain Code

Size of scene[328]

edit scene size via code[329]

black area at editing scene size[330]

extend borders of scene[331]

Terrain borders.[332]

Flora[333]

hard-coded trees at scene[334]

Biggest possible scene[335]

5.4.2.5 Outer Terrain Border

eleventh field is missing, optional in some unusual way.

[323] Ruthven (credit) and kalarhan, Modding Q&A and Modding Q&A.
[324] Swyter, Modding Q&A, and GetAssista, Modding Q&A.
[325] InVain, Mount & Blade Modding Discord.
[326] InVain, Modding Q&A, Mount & Blade Modding Discord and Mount & Blade Modding Discord.
[327] Yoshiboy, Questions about making new rooms and some other questions.
[328] Vornne, Modding Q&A.
[329] kalarhan, Modding Q&A.
[330] Lumos, Modding Q&A and Modding Q&A.
[331] InVain, Modding Q&A.
[332] Modding Q&A.
[333] Dargor, Modding Q&A.
[334] jacobhinds, Modding Q&A.
[335] InVain, Mount & Blade Modding Discord.

5.4.2.6 Other Notes

scene stuff[336]

5.4.2.7 Character Creation Screen aka Meeting Scene

A first look around would perhaps let you think that it is the scene *meeting_scene_plain*. It is however no scene, it is made up by the game engine on the fly. It doesn't even have a real terrain, the game engine is just loading the mesh `ch_meet_plain_a` (CommonRes/particles_2.brf). Note that you can not change the lighting settings at it.[337] The character creation scene uses the very first skybox defined in Skyboxes.py.[338] The character uses the animation `stand_man` which is hardcoded.[339]

[336] cmpxchg8b, Modding Q&A.

[337] cmpxchg8b, Modding Q&A, Modding Q&A and Modding Q&A.

[338] _Sebastian_, Mount & Blade Modding Discord.

[339] Veledentella, Mount & Blade Modding Discord.

5.4.3 Module Mission Templates

```
Each mission-template is a tuple that contains the following fields:
1) Mission-template id (string): used for referencing mission-templates in other files.
   The prefix mt_ is automatically added before each mission-template id
2) Mission-template flags (int): See header_mission_templates.py for a list of available flags
3) Mission-type(int): Which mission types this mission template matches.
   For mission-types to be used with the default party-meeting system,
   this should be 'charge' or 'charge_with_ally' otherwise must be -1.
4) Mission description text (string).
5) List of spawn records (list): Each spawn record is a tuple that contains the following fields:
  5.1) entry-no: Troops spawned from this spawn record will use this entry
  5.2) spawn flags.
  5.3) alter flags. which equipment will be overriden
  5.4) ai flags.
  5.5) Number of troops to spawn.
  5.6) list of equipment to add to troops spawned from here (maximum 8).
6) List of triggers (list).
   See module_triggers.py for infomation about triggers.
```

The file **module_mission_templates.py** contains the Mission Templates. Mission templates are sets of scripts that govern different events known as missions, in simple terms. They can range from simple quests to complex battles with reinforcements.[340] If you need to test outcomes of combat, this is the file you will need to work in.

Anytime you have a confrontation battle, these code blocks are checked. You can also set up what macro keys (such as TAB) will do during the battle. Before the meat of the mission templates, a bunch of local constants are set. The first two are designed as item overrides. We will see more on this later, but if you are picking up the workings of the Module System, you should understand what they do. Next you will see some other longer tuple constants, such as common_battle_mission_start. You should recognize the format of these constants. They look just like tuples from module_triggers.py. This is because they are triggers. As you can see in the header of module_mission_templates.py, the sixth part of these tuples are triggers. By defining some common triggers at the top as constants, you will not have to type these triggers for each tuple that will need them.

At about line 1275, you will see the start of the mission templates `mission_templates = [` . Let's look at the breakdown of the first mission template tuple "town_default" as defined in this file's header:

[340] Veledentella, What are mission templates.

```
1. Mission template ID (prefixed with mt_) = "town_default"
2. Flags (as defined in header_mission_template.py) = 0
3. Mission Type interger. Should be 'charge' or 'charge_with_ally' otherwise must be -1 = -1
4. Mission description text = "Default town visit"
5. List of spawn records (list). Breakdown of the list (entry-no spawn point, spawn flags, alter/equipment override flags, AI flags, number of troops to spawn, list of equipment to add to troops spawned here with a maximum of eight items) =
 [
 (0,mtef_scene_source|mtef_team_0,af_override_horse,0,1,pilgrim_disguise),
...
 (7,mtef_scene_source|mtef_team_0,af_override_horse,0,1,[]),
 (8,mtef_scene_source,af_override_horse,0,1,[]),
...
 (12,mtef_scene_source,af_override_horse,0,1,[]),
 (13,mtef_scene_source,0,0,1,[]),
 (14,mtef_scene_source,0,0,1,[]),
 (15,mtef_scene_source,0,0,1,[]),
 (16,mtef_visitor_source,af_override_horse,0,1,[]),
...
 (31,mtef_visitor_source,af_override_horse,0,1,[]),
 ],
6. List of triggers (see module_triggers.py for format on triggers) =
 [
 (1, 0, ti_once, [], [
    (store_current_scene, ":cur_scene"),
    (scene_set_slot, ":cur_scene", slot_scene_visited, 1),
    (try_begin),
        (eq, "$sneaked_into_town", 1),
        (call_script, "script_music_set_situation_with_culture", mtf_sit_town_infiltrate),
  (else_try),
        (eq, "$talk_context", tc_tavern_talk),
        (call_script, "script_music_set_situation_with_culture", mtf_sit_tavern),
  (else_try),
        (call_script, "script_music_set_situation_with_culture", mtf_sit_travel),
  (try_end),
 ]),
 (ti_before_mission_start, 0, 0, [], [(call_script, "script_change_banners_and_chest")]),
 (ti_inventory_key_pressed, 0, 0, [(set_trigger_result,1)], []),
 (ti_tab_pressed, 0, 0, [(set_trigger_result,1)], []),
 ],
),
```

5.4.3.1 Mission Types

cancel_attack

cancel_reinforce

charge

charge_with_ally

intend_battle

leave_during_battle

leave_wo_battle

speak

stay_back

stay_back_with_ally

surrender

5.4.3.2 Spawn Records

From Mirathei, need to explain it better. Also note of Khamu about mission template entry and spawn record[341]

```
[  ( 1 ,mtef_ defenders ,0, group(1)|aif_start_alarmed  ,8,[]),
     ^ Entry number
   ( 0 , mtef_defenders ,0, group(1)|aif_start_alarmed  ,0,[]),
               ^ Who spawns here
   ( 4 , mtef_attackers ,0, aif_start_alarmed  ,8,[]),
                                     ^ How do they behave
   ( 4 , mtef_attackers ,0, aif_start_alarmed  ,0,[]),
                                                ^ How many spawn
],
```

[341] Khamukkamu, Mount & Blade Modding Discord.

5.4.3.3 Mission Template Spawn/Entry Flags

After the spawn entry points, you need to set the spawn/entry flags. The first flag sets the confrontation style, also called 'filter flags' in header_mission_templates.py. This can be for example mtef_attackers or mtef_defenders, the flags for field encounters. Attackers would be the party initiating the encounter, the defender would be the other party. Multiple flags are divided by a '|'. All available flags are given in the list below:

mtef_ally_party not used in Native, up4research.

mtef_archers_first gives archers troops of the party priority at spawning as first ones at this entry point(?).

mtef_attackers sets the spawned troop to be part of the party which initiated the encounter(?).

mtef_cavalry_first gives cavalry troops of the party priority at spawning as first ones at this entry point(?).

mtef_conversation_source not used in Native, up4research.

mtef_defenders sets the spawned troop to be part of the party which got encountered(?).

mtef_enemy_party not used in Native, up4research.

mtef_infantry_first gives infantry troops of the party priority at spawning as first ones at this entry point(?).

mtef_leader_only bundels `mtef_no_companions` and `mtef_no_regulars` so that only the leader of a party spawns at this entry point(?).

mtef_no_auto_reset if you want to move entry points you have to mark them as mtef_no_auto_reset. Otherwise they will "auto reset" every ten frames.[342]

mtef_no_companions prevents companions of parties to spawn at this entry point(?).

mtef_no_leader prevents leaders of parties to spawn at this entry point(?).

mtef_no_regulars prevents regulars of parties to spawn at this entry point(?).

mtef_regulars_only bundels `mtef_no_leader` and `mtef_no_companions` so that only regulars of a party spawn at this entry point(?).

[342] cmpxchg8b, Modding Q&A.

mtef_reverse_order spawns troops in reverse order regarding to their sorting at the party(?).

mtef_scene_source is used for troops that have a site/entry point defined statically (see tuple field 5 at module_troops.py). It spawns them on the first frame of the mission only if the entry point defined in the tuple has the flag.[343]

mtef_team_0 sets that the spawned troop would be part of team 0.

mtef_team_1 sets that the spawned troop would be part of team 1.

mtef_team_2 sets that the spawned troop would be part of team 2.

mtef_team_3 sets that the spawned troop would be part of team 3.

mtef_team_4 sets that the spawned troop would be part of team 4.

mtef_team_5 sets that the spawned troop would be part of team 5.

mtef_team_member_2 not used in Native, up4research.

mtef_use_exact_number lets the exact number of troops spawn here which are declared at a later part of the spawn record.

mtef_visitor_source is used for adding visitors to the site on the fly. It enables the usage of the operation `add_visitors_to_current_scene` at this entry point.[344]

5.4.3.4 Mission Template Alter Flags

We now look at the alter flags. These are set to override certain items that may not be allowed in this mission. If it would be more sporting to fight on foot at your planned mission, you will want to override horses. All available flags are given in the list below:

af_override_weapons overrides all weapons.

af_override_weapon_0 overrides the weapon in equipment slot 0.

af_override_weapon_1 overrides the weapon in equipment slot 1.

af_override_weapon_2 overrides the weapon in equipment slot 2.

af_override_weapon_3 overrides the weapon in equipment slot 3.

[343] cmpxchg8b, Modding Q&A.

[344] cmpxchg8b, Modding Q&A.

af_override_head overrides the helmet in equipment slot 4.

af_override_fullhelm overrides the helmet in equipment slot 4 if the item entry of the helmet has the flag `itp_covers_head`.[345]

af_override_body overrides the armour in equipment slot 5.

af_override_foot overrides the footwear in equipment slot 6.

af_override_gloves overrides the gloves in equipment slot 7.

af_override_horse overrides the horse in equipment slot 8.

af_override_everything bundles some alter flags such way that everything gets overriden.

af_override_all bundles some alter flags such way that everything except the footwear gets overriden.

af_override_all_but_horse bundles some alter flags such way that everything except the footwear and the the horse gets overriden.

af_require_civilian lets the troop only wear items with the flag `itp_civilian`.

The next flag is the AI flag. A look into module_mission_templates.oy shows you that in the most cases it is set to aif_start_alarmed which makes the AI ready to fight. Otherwise it is set to 0. After this you set the number of troops to spawn.

The last part of the list (inside the '[]') is for equipment to add to the troops spawned here, to a maximum of 8 items. If you override all other equipment the troop will pick from what is listed here. If you want the troop spawning here to have a choice between a wooden staff or a wooden sword for example, override all equipment and add the aforementioned items to be randomly chosen.

[345] Somebody, Modding Q&A.

5.4.3.5 Mission Template Triggers[346]

simple triggers, triggers and mission templates explained, taking over the other passages to the upfollowing chapters.[347]

A trigger in mission templates (and in module_triggers for that matter) takes the following form
First Part: how often the trigger is checked, either in seconds* or as a hard-coded event such as "ti_on_agent_hit"
Second part: the delay, in seconds*, between reading the conditions block and executing the consequences block. This does not work if you are using a “word“/hard-coded event interval, such as "ti_on_agent_hit"
Third part: the re-arm delay, in seconds*, between when the consequences block was finished and when the trigger can be checked again. A special re-arm delay "ti_once" is used for triggers that should only complete themselves (have their condition pass and then execute their consequence) once and then never fire again.
Fourth part: the conditions block, always executed when the trigger is called
Fifth part: the consequences block, only executed if the conditions block does not fail, and after the delay interval has passed.
*Note, unlike other numbers in the module system, the 3 intervals/delays of triggers do NOT need to be integers.
In practice, it looks like this:

```
(10, 2, 60,
  [
   (eq, 1, 1),
  ],
  [
   (val_add, "$global_variable", 10),
  ]),
```

In this instance, the check interval is every 10 seconds, so the trigger will be checked every 10 seconds. The delay interval is 2 seconds, so if the conditions block is true, the consequences block will fire 2 seconds later. The re-arm interval is 60 seconds, so after the conditions block is found to be true, it will be one minute until this trigger can be checked again.

The conditions block is a simple equality test 1==1, that will always pass, so the consequences block will always fire 2 seconds after the trigger is checked. The consequences block then takes a global variable and increments it by 10.

[346] Taken from *An Introduction to Module Syntax and Usage* by Caba‘drin
[347] Lumos, Modding Q&A.

Timed Triggers

Understanding the check interval/delay/re-arm intervals of timed triggers can be tricky so here is an illustration:

```
(2, 1, 3, [<a condition that sometimes fails, sometimes passes>],
    [
     #some consequences
    ]),
```

This trigger, like all timed triggers with a check interval >0, will start to be checked around the first 1 second of the mission:

```
Second      Event
1              Trigger checked; condition fails—apply check interval (+2 seconds)
3              Trigger checked; condition fails—apply check interval (+2 seconds)
5              Trigger checked; condition passes—apply consequence delay (+1 second)
6              Consequence fires and completes—apply check interval (+2); apply re-arm delay (+3)
11             Trigger checked; condition fails—apply check interval (+2 seconds )
13             Trigger checked; condition passes—apply consequence delay (+1 second)
14             Consequence fires and completes—apply check interval (+2); apply re-arm delay (+3)
19             Trigger checked....
```

So, although we have specified a “check interval“ of 2 seconds, we see that the trigger is not checked only on even seconds; instead it is checked in seconds 1, 3, 5, 11, 13, 19.

If one wants a completely “scheduled“ trigger, both consequence and re-arm delays cannot be used.

Two Triggers of the Same Type/Interval

When there are two triggers of that have the same check interval (be that in seconds, or on an event ti_*) the order they appear in the mission template matters. The trigger that appears first in the template will fire first, followed by the next trigger of the same interval, and so on. That means if you have two triggers that fire ti_on_agent_spawn, the one that appears in the file first will execute before the second one.

Triggers near the beginning of a Mission

Logically, the trigger ti_before_mission_start takes place before the scene is set up and before agents are spawned into the scene.[348] Next, spawning takes place before any other triggers fire– ti_on_agent_spawn triggers are the only triggers firing at this point. Next, ti_after_mission_start triggers fire, as well as any triggers with a check interval of 0 (every frame) as the mission timer starts. Event-based triggers will not be called until their ti_* event occurs; other timed triggers begin being called somewhere in the first second of the mission, though after ti_after_mission_start triggers and every frame (check interval 0) triggers.

If you have triggers that do not need to fire that close to mission start, adding something like

```
(store_mission_timer_a, reg0),(gt, reg0, <second-to-start-checking>),
```

to the conditions block will help ease the CPU load at mission start.

Neither ti_before_mission_start nor ti_after_mission start need "ti_once" in their re-arm delay as they will only ever fire one time.

To summarize, at the start of a mission we have:

- ti_before_mission_start
- Spawning– (ti_on_agent_spawn)
- Time Begins–
- ti_after_mission_start & triggers with check intervals of "0" (which fire at a mission time of approximately 0.2 seconds)
- Timed Triggers (still within the first 1 second of the mission)
- Event-based triggers

ti on item wielded is mp only[349] Nah, I saw it in singleplayer mission templates too.

The problem is, for throwing weapons and with the ti_on_agent_hit trigger, the weapon can will fly through people and hit multiple people.[350] As always, one modder's bug can be another modder's feature.

equivalent triggers[351]

rearms and delays at named triggers[352] no delay and no rearm for specific triggers[353] timed triggers and event triggers (named triggers)[354]

[348] It's hard to spawn something in a scene if the scene itself doesn't exist yet; cmpxchg8b, Modding Q&A.

[349] Somebody, Modding Q&A.

[350] SonKidd, Modding Q&A.

[351] Caba'drin, Modding Q&A.

[352] cmpxchg8b, Modding Q&A.

[353] Somebody, Modding Q&A, and kalarhan, Modding Q&A.

[354] Vornne, Modding Q&A.

trigger check interval[355]

5.4.3.6 Jumping from Mission Template to Mission Template[356]

There might come up a situation at your modding project at which you would like to switch from a scene to another scene without any menu between. This could be because of the plot or story driven basics of your mod or because you want to design a special scenario appears in the player's campaign. Now you cannot make a simply jump at the end of the mission template since after the mission terminates, the game will send the players to the game menu that was first loaded when they encountered the map party. For example if it was a town, they get booted back to the town menu. At that point you could cause a new scene to be loaded.

The solution here is however to basically just move the "jump_mission"-operations to the conditions block. Then the players can jump straight from one mission to another. For this, change your triggers to look similar to the following example:

```
    (1, 60, ti_once,
    [
      (store_mission_timer_a,reg(1)),
      (ge,reg(1),10),
      (all_enemies_defeated,2),
      (neg|main_hero_fallen),
      (set_mission_result,1),
      (assign,"$battle_won",1)
      (set_jump_mission,"mt_force_conversation"),
      (assign,"$other_char","$other_char_win"),
      (jump_to_scene,"$other_scene_win"),
      (change_screen_mission),
      ],[]),

      (1, 4, ti_once,
    [
      (main_hero_fallen)
      (set_jump_mission,"mt_force_conversation"),
      (assign,"$other_char","$other_char_lose"),
      (jump_to_scene,"$other_scene_lose"),
      (assign,"$player_lost",1),
      (change_screen_mission),
    ],[]),
```

[355] kalarhan, Modding Q&A.

[356] Based upon a discussion between fisheye, ex_ottoyuhr and silverkatana at the Modding Q&A.

5.4.3.7 Weather

Some weather info bits[357]

random rain at scene[358]

It can’t rain and snow at the same time[359]

multiplayer weather[360]

crossbows shoot bad at fog via engine?[361]

basic principles of using weather in a scene[362]

yellow fog instead of black[363]

low fog value probably due to yellow fog problem?[364]

weather stuff[365]

change skybox for scene (not sure if weather or mission template)[366]

paint skybox[367]

mp weather stuff[368]

weather in edit mode just for preview[369]

setting weather[370]

remove fog from scene[371]

Problems with fog[372]

global cloud amount sets fog too[373]

skyboxes daytime[374]

Darker nights[375]

Global cloud amount[376]

[357] Caba‘drin, Modding Q&A.
[358] Roemerboy, Modding Q&A.
[359] Somebody, Modding Q&A.
[360] Somebody, Modding Q&A
[361] Somebody, Modding Q&A and Modding Q&A.
[362] Abhuva, Nordous’ Sceners Guild.
[363] Somebody, Modding Q&A.
[364] Marko, Modding Q&A.
[365] Sim00n (credit), Modding Q&A.
[366] kalarhan, Modding Q&A.
[367] kalarhan, Modding Q&A.
[368] Namakan (credit), Modding Q&A, and _Sebastian_, Modding Q&A.
[369] _Sebastian_, Modding Q&A.
[370] _Sebastian_, Modding Q&A.
[371] Dj_FRedy, Modding Q&A.
[372] here and here.
[373] Khamukkamu, Modding Q&A.
[374] Ruthven, Modding Q&A.
[375] Pitch, Modding Q&A.
[376] cmpxchg8b, Modding Q&A.

5.4.3.8 Other notes

Maximum number of entry points per scene is 128, ranging from 0 to 127. You can however use some unique scene props that could act as some sort of makeshift entry points.[377]

Formula used to calculate the number of agents a player gets at a battle's start.[378]

ti_on_agent_dismount.[379] ti on agent dismount fires when horse dies [380], so at both[381]

add_visitors_to_current_scene and entry_point_set_position.[382]

Teams in Singleplayer[383]

Condition block check timing.[384]

mtf_battle_mode[385]

ti_on_leave_area and ti_on_player_exit[386]

Delay doesn't work for named triggers but rearm works fine[387]

Teams count SP and MP.[388] InVain also made some little test there, need to fetch note for that.

ti_on_agent_killed_or_wounded[389]

Bundle triggers (read comments along the whole page)[390]

Teams in SP[391]

Adding list of triggers[392]

Adding triggers to Mission template always at top.[393]

maximum number teams in mission[394]

ti on item dropped only mp[395]

entry point limit[396]

Some stuff explained already, check if all present[397]

[377] Lumos, Warband Script Enhancer v3.2.0, and cmpxchg8b, Modding Q&A, Modding Q&A, Modding Q&A and additional remark at Modding Q&A.
[378] Caba'drin, Modding Q&A.
[379] Caba'drin and dunde, Modding Q&A.
[380] Caba'drin and Mammoet, Modding Q&A.
[381] Caba'drin, Modding Q&A.
[382] Vornne and dunde, Modding Q&A.
[383] Caba'drin, Modding Q&A.
[384] GetAssista, Modding Q&A.
[385] cmpxchg8b, Modding Q&A.
[386] Caba'drin and Somebody, Modding Q&A.
[387] cmpxchg8b, Modding Q&A and Modding Q&A.
[388] Somebody, Modding Q&A.
[389] Caba'drin, Modding Q&A.
[390] Caba'drin, Modding Q&A.
[391] Caba'drin, Modding Q&A.
[392] Caba'drin, Modding Q&A.
[393] _Sebastian_, Modding Q&A.
[394] dunde and Caba'drin, Modding Q&A.
[395] dunde, Modding Q&A.
[396] Lumos, Modding Q&A.
[397] Somebody, Modding Q&A.

strange ai behaviour[398]

no consequences after trigger[399]

open questions about spawning agent[400]

entry points and entry number[401]

mt flags override player inventory[402]

mp end map[403]

how much time is it (add also to module triggers if not already)[404]

listing triggers[405]

bundle triggers[406]

some mission template stuff[407]

no delays for engine events[408]

hard-coded mission templates[409]

list of triggers[410]

entry points at mission templates[411]

additional teams remark[412]

prevent save or quicksave mid-scene[413]

entry point vs spawn record[414]

Characters remove helmets in conversations[415]

Override behaviour of some keys[416]

list of triggers[417]

scene_source flag[418]

[398] Multiple discussions.

[399] Swyter, Modding Q&A.

[400] Lav, Modding Q&A.

[401] Waldzios (credit), Modding Q&A.

[402] Lav, Modding Q&A.

[403] Ikaguia, Modding Q&A.

[404] The_dragon, Modding Q&A.

[405] Somebody, Modding Q&A.

[406] kalarhan, Modding Q&A.

[407] kalarhan, Modding Q&A.

[408] _Sebastian_, Modding Q&A.

[409] kalarhan, Modding Q&A.

[410] kalarhan, Modding Q&A.

[411] Khamukkamu and kalarhan, Modding Q&A.

[412] Vornne, Modding Q&A.

[413] kalarhan, Modding Q&A.

[414] kalarhan, Modding Q&A.

[415] kalarhan, Modding Q&A.

[416] kalarhan, Modding Q&A.

[417] kalarhan, Modding Q&A.

[418] need to reread, Strange "set_visitor" error.

charge flag[419]

Scene source people never wear hats?[420]

Entry points limit[421]

Working around 127 event limit[422]

[419] nijis and bryce777 (credit), Breaking the random encounter scene spawnpoints.
[420] Yoshiboy, NPCs not wearing hats.
[421] cmpxchg8b, [WB] Warband Script Enhancer v3.2.0.
[422] cmpxchg8b, [WB] Warband Script Enhancer v3.2.0.

5.4.4 Flora Kinds

Compiling the file[423]

compiling data files[424]

5.4.5 Skyboxes

format skyboxes, [425]

Also, is it possible to alter the worldmap skybox so it changes with the time like in battles?

It is kind of possible, you have to create a party with the sky box for the mesh and have a trigger that constantly makes it re-locate to the player party. The problem with this is that it disappears when the player party is not in view. Essentially going back to normal. You can then switch this party for different times of day.

Last Question at thread here

[423] kalarhan, Modding Q&A.

[424] Dj_FRedy, Modding Q&A.

[425] Yoshiboy, Data folder modular.

5.5 Map-related Module System Files

5.5.1 Module Map Icons

```
Each map icon record contains the following fields:
1) Map icon id: used for referencing map icons in other files.
   The prefix icon_ is automatically added before each map icon id.
2) Map icon flags. See header_map icons.py for a list of available flags
3) Mesh name, for Native they can be found in the brf files map_icon_meshes,
    map_icons_b, and map_icons_c
4) Scale.
5) Sound.
6) Offset x position for the flag icon.
7) Offset y position for the flag icon.
8) Offset z position for the flag icon.
```

module_map_icons.py begins with two constants that are used to govern the scale of the map 'avatars' and banners which we see in-game.

```
banner_scale = 0.3
avatar_scale = 0.15
```

They are pretty much self-explanatory. Editing the value at banner_scale rescales the size of the banners and the value at avatar_scale accordingly the size of the 'avatars' at the world map. You will see those constants getting used at the entries of the map icons, to avoid making accidentally some misstake at copy pasting the numbers. You are free to add new constants there, fitting to your purposes. Hereinafter you will find the following lines:

```
map_icons = [
("player", 0, "player", avatar_scale, snd_footstep_grass, 0.15, 0.173, 0),
("player_horseman", 0, "player_horseman", avatar_scale, snd_gallop, 0.15, 0.173,
    0),
("gray_knight", 0, "knight_a", avatar_scale, snd_gallop, 0.15, 0.173, 0),
("vaegir_knight", 0, "knight_b", avatar_scale, snd_gallop, 0.15, 0.173, 0),
...
```

Taking again the first tuple ("player",0,"player", avatar_scale, snd_footstep_grass, 0.15, 0.173, 0), as example of observation:

```
1) name of the icon = "player"
2) icon flags = 0
3) Mesh name = "player"
4) Mesh scale = avatar_scale
```

```
5) sound id = snd_footstep_grass
6) Flag offset x = 0.15
7) Flag offset y = 0.173
6) Flag offset z = 0
```

The player icon has no icon flag and is using the mesh “player“ as visual. The mesh is scaled down to the constant avatar_scale which has been defined before as 15 percent. While the icon is moving on the world map the sound footstep_grass is getting played.

Take note that sounds entered in tuple field 5) need to be set up as well in module_sounds.py. Be also aware that multi-meshing is not possible at map icons. You can however have multiple static parties in one location (like bridges), with only one of them being e.g. the city itself.[426]

There is a limit of 256 map icons hardcoded into the game engine. Most modders get around this by condensing the hundreds of flag icons into heraldic ones.[427] While the limit of 256 map icons is getting applied by the operation `party_set_icon` too, modders who seek an 256+ range can also make use of the operations `party_set_banner_icon` and `party_set_extra_icon` which are also working for 256+ map icons.[428]

5.5.1.1 Map Icon Flags

mcn_no_shadow prevents that the respective map icon casts a shadow.

5.5.1.2 Map Icon Trigger

ti_on_init_map_icon will run if a map icon is initialized. Trigger Parameter 1 is the ID of the Party whose map icon was just initialized.

[426] Multiple people, Modding Q&A.

[427] Somebody, Modding Q&A. Of good use here is the FISH&CHIP OSP - Heraldry&Retexturing Unleashed by Lav.

[428] Burspa, Mount & Blade Modding Discord.

5.5.2 Module Parties

```
Each party record contains the following fields:
 1) Party id: used for referencing parties in other files.
    The prefix p_ is automatically added before each party id.
 2) Party name. This is the party's name as it will appear in-game. Can be as different from the party-id as you like.
 3) Party flags. See header_parties.py for a list of available flags. The first flag of every party object must be the icon that this party will use given that you have not set the flag pf_disabled.
 4) Menu. ID of the menu to use when this party is met. The value 0 (the constant no_menu is 0) uses the default party encounter system.
 5) Party-template. ID of the party template this party belongs to. Use pt_none as the default value.
 6) Faction. This can be any entry from module_factions.py.
 7) Personality. See header_parties.py for an explanation of personality flags.
 8) AI-behavior. How the AI party will act on the world map.
 9) AI-target party. The AI-behaviour's target.
10) Initial coordinates. The party's starting coordinates on the overland map; X, Y.
11) List of stacks. Each stack record is a triple that contains the following fields:
  11.1) Troop-id. This can be any regular or hero troop from module_troops.py.
  11.2) Number of troops in this stack; does not vary. The number you input here is the number of troops the town will have.
  11.3) Member flags. Optional. Use pmf_is_prisoner to note that this member is a prisoner.
    Note: There can be at most 6 stacks.
12) Party direction in degrees [optional]
```

At the beginning of the file **module_parties.py** you will see some constants declarations. They can make repeated flags a lot easier to enter. You are again free to add there your own constants.

```
no_menu = 0
#pf_town = pf_is_static|pf_always_visible|pf_hide_defenders|pf_show_faction
pf_town = pf_is_static|pf_always_visible|pf_show_faction|pf_label_large
pf_castle = pf_is_static|pf_always_visible|pf_show_faction|pf_label_medium
pf_village = pf_is_static|pf_always_visible|pf_hide_defenders|pf_label_small
```

First, Understand that parties can be anything that you come in contact with on the map. These can be traveling groups or stationary villiages, towns or other locations of your own design. One key to parties is that an encounter is triggered when two or more parties meet. In general, static parties are those found in module_parties.py, like the usual towns, castles and villages on Native and dynamic parties are those created using a party template in party_templates.py. The static ones can be enabled and disabled - think of a town that is visible as active. They all have a known ID. The dynamic parties on the other hand can be created and destroyed. Every new spawned or created party gets a new ID as an unique identifier. Thus you need to check if a certain ID is still valid, controlling for if that dynamic party is still around or was already destroyed.

Let's look at an example of a party. If you scroll down a bit you will come to the tuple

```
("town_1","Sargoth", icon_town|pf_town, no_menu, pt_none, fac_neutral,0,
    ai_bhvr_hold,0,(-1.55, 66.45),[], 170),
```

This tuple places the town of Sargoth on the map. Sargoth's various qualities are set in the appropriate fields which can be broken down as follows:

```
1) Party-id = "town_1"
2) Party name = "Sargoth"
3) Party flags = icon_town|pf_town
4) Menu = no_menu
5) Party-template = pt_none
6) Party faction = fac_neutral
7) Party personality = 0
8) AI-behaviour = ai_bhvr_hold
9) AI-target party = 0
10) Initial coordinates = (-1.55, 66.45)
11) List of troop stacks = [] (None)
12) Party direction = 170
```

The field 4) is deprecated, which means that it's outdated and no longer used. As of M&B version 0.730, this field has no effect whatsoever in the game. Instead the script game_event_party_encounter is getting called from the game engine whenever the player party encounters another party or a battle on the world map. The same applies at field 4) in module_party_templates.py.

When adding an own town, castle or village take note: In the case of module_parties.py and certain other module files, you should not add your new towns at the bottom of the list. There will be comments in these files warning you about doing this, as it can break operations in the native code. In module_parties.py, it is recommended that you add for example any new towns between "town_1" and "castle_1". This is defined in the module_constants.py. Look up there the defined ranges for the different categories of map icons. Towns, villages and castles get usually added as neutral since they are getting assigned to a faction and a lord at game_start.

5.5.2.1 Checking of Coordinates at World Map

The 10th field of the tuple requires valid coordinates from your side. Ingame, with Edit Mode on and “Show framerate“ ticked from the launcher, you will get the coordinates of the player’s party (`"p_main_party"`) displayed at the top-left corner when pressing Ctrl+E.[429] The coordinates end up in the top left in green colour right under the framerate counter. If you don’t have a “Terrain“ button at the bottom left next to “Camp“ you are probably not in Edit Mode.[430] Turn on cheat mode, too, and ctrl-click to any point on the map to get the coordinates.

An alternative is to use of the operation (party_get_position) as follows:

```
(set_fixed_point_multiplier, <precision>),
(party_get_position, pos1, "p_main_party"),
(position_get_x, reg1, pos1),
(position_get_y, reg2, pos1),
(display_message, "@Player is at ({reg1}, {reg2})"),
```

Use whatever value you want as <precision> - popular values are 100 and 1000. If you use 1, then you are using the base distance units. On the global map that is the distance that a party with speed of 1.0 traverses in 1 second.[431] Another scripting approach is the following[432]:

```
cmenu_locate = 10
```

```
        ("game_context_menu_get_buttons",
          [
                (store_script_param, ":party_no", 1),
                (context_menu_add_item, "@Where am I?", cmenu_locate),
                (try_begin),
                  (neq, ":party_no", "p_main_party"),
                  (context_menu_add_item, "@Move here", cmenu_move),
                (try_end),
...
        ("game_event_context_menu_button_clicked",
          [ (store_script_param, ":party_no", 1),
                (store_script_param, ":button_value", 2),
                (try_begin),
                  (eq, ":button_value", cmenu_center_note),
                  (change_screen_notes, 3, ":party_no"),
                (else_try),
                  (eq, ":button_value", cmenu_locate),
```

[429] Caba‘drin, Modding Q&A, and Ikaguia, Modding Q&A.
[430] kt0, Modding Q&A.
[431] Lav, Modding Q&A.
[432] Somebody, Modding Q&A.

```
                    (party_get_position, pos1, ":party_no"),
                    (str_store_party_name, s1, ":party_no"),
                    (position_get_x, reg1, pos1),
                    (position_get_y, reg2, pos1),
                    (display_message, "@{s1} is at x:{reg1}, y:{reg2}"),
```

One more way to get a location on the map is to hold down the Z key in Thorgrim's Map Editor, and move the cursor over the map. When you release the z key the location is also copied into the clipboard so you can paste it into a python script or whatever you need it for.[433] It is also possible to just make all changes in the map editor, save it so it updates the parties.txt, and then use either HokieBT's UpdateModuleParties or kt0's Party Script to copy the coordinates back to your python code.[434]

5.5.2.2 Party Flags

pf_disabled hides the party at the world map, used for bandit spawnpoints or temporary parties for example. Locations still show up in the factions list though. Scripts and `try_for_parties` loops still see it.

pf_is_ship enables the party to travel on water. Passable terrains are: rt_water, rt_river and rt_bridge.

pf_is_static marks a party which will not move, like locations for example.

pf_label_small assigns a small label to that party, used for villages for example.

pf_label_medium assigns a medium label to that party, used for castles for example.

pf_label_large assigns a large label to that party, used for towns for example.

pf_always_visible makes villages, castles and towns always visible on the worldmap (to the player sight, AI parties won't bother the flag)[435]. Otherwise the player only sees that party within spotting range.

pf_default_behavior (?) unknown, up for research.

pf_auto_remove_in_town disbands travelling parties after reaching a town. Used for caravans, a group of survivors after a battle (running away to a secure location/routed), a messenger, etc.

[433] Thorgrim, Modding Q&A with reference to Thorgrim's Map Editor.

[434] HokieBT, Modding Q&A.

[435] dunde, Modding Q&A.

pf_quest_party used for the player party main_party via scripts. Setting it to 1 tells the game that the player is on a quest. You can set it back to 0 if the quest gets canceled. Used for the tutorial and new game quest chain. (Not sure if still up to date)

pf_no_label hides the party name at the world map.

pf_limit_members marks that the party size is restricted according to the leadership skill of the leader of the party. In Native only used for the player party. This flag works for any party and you can remove this from player party to disable the cap. The operation add_companion_party sets this flag to a spawned party. The flag will get ignored when using the operations party_force_add_members and party_force_add_prisoners.

pf_hide_defenders hides the garrison numbers on the world map.

pf_show_faction shows the faction name of a party, without this flag the party will appear like looters and manhunters.

pf_is_hidden commented out flag; used in the engine, do not overwrite this flag!

pf_dont_attack_civilians prevents that this party can attack parties with the flag pf_civilian.

pf_civilian marks a party as civilian so that parties with the flag pf_dont_attack_civilians are not attacking it.

pmf_is_prisoner is a party member flag which turns the selected troop into a prisoner when getting added to the party via entry field 12).

carries_goods(x) used for slowing down the party, the higher x the bigger the slow down. The factor x gets multiplied with the factor 50, the result gives the additional weight the party has to carry. This flag does not cause additional loot. The engine contains following formula:

```
float totalWeight = ((m_flags & pf_carry_goods_mask) >> pf_carry_goods_bits
    ) * 50.0f;
```

Not sure if and how useable the two upfollowing flags are. They are most probably unused by the engine.

pf_carry_gold_multiplier defines the multiplicator effect for carries_gold(x).

carries_gold(x) unused by the engine.

5.5.2.3 Party Speed[436]

Party speed is how fast a party moves on the map. It is affected by many factors, including morale, the party's highest Pathfinding skill, the amount of units in a party which are mounted, prisoners, the terrain, day and night cycle, inventory items (e.g., iron) and weather. Some troops can be marked with the flag tf_mounted which enables that these troops' movement speeds on the map are determined by their riding skill even though they fight on foot, such as Hired Blades. Morale is one of the simplest factors in the equation, as it can be increased through the Leadership skill, food variety, a smaller party size, and positive recent events such as winning battles.

Weight accumulated from inventory items, particularly trade goods[437], can significantly slow a party down. This is also where the flag carries_goods(x) gets important. Some of the encumbering effects of trade goods can be counteracted by having spare horses in the inventory which will act as pack animals and carry the trade goods for the player. The type of horse is irrelevant. However, horses themselves also slow you down, so filling your inventory with horses will result in a negative effect. The speed penalty for larger party sizes is also unaffected by extra horses in the inventory.

For the exact formule check subchapter 7.3.1, World Map Speed Formula.

5.5.2.4 AI-Behaviour Settings

ai_bhvr_hold lets a party stand completely still on the world map, given that the party's faction is neutral with every other faction.

ai_bhvr_travel_to_party gives the party a predetermined travel destination. The party doesn't disappear when reaching its destination (just like caravans don't) but turns invisible, and there is no party encounter you can pick up in code or dialogs.

ai_bhvr_patrol_location lets a party patrol around a predetermined area, like Steppe Bandits in Native.

ai_bhvr_patrol_party lets a party patrol around a predetermined other party, like Steppe Bandits in Native.

ai_bhvr_track_party deprecated, use the alias ai_bhvr_attack_party instead.

ai_bhvr_attack_party lets a party attack a predetermined other party.

ai_bhvr_avoid_party lets a party avoid a predetermined other party.

[436] Working in the informations given in the thread *Map speed/pack animals, Trade skill and town relations tested* by Sicnarf and for some clear descriptions from the page *Party speed* of the Mount&Blade Wiki Fandom.

[437] The weight of weapons and armouries has appearantly no effect on the party speed.

ai_bhvr_travel_to_point gives the party a predetermined position on the world map as travel destination.

ai_bhvr_negotiate_party (?) unknown, up4research.

ai_bhvr_in_town gives the information that a party is in a town.

ai_bhvr_travel_to_ship (?) unknown, up4research. Gives the party a predetermined travel destination which is a ship?

ai_bhvr_escort_party lets the party escort a predetermined other party.

ai_bhvr_driven_by_party drives this party by a predetermined other party. Used in Native for the cattle herd quest.

5.5.2.5 Maximum Number of Stacks[438]

While there can be at most 6 stacks in the party tuple you can add up to 255 stacks via scripts after template is spawned via the operations `party_add_members`, `party_add_prisoners` or the forced versions of them. if you open the party screen with `change_screen` operations you will have a scrollable list but it obviously wouldn't fit in global map tooltip. There you see at maximum 32 party member stacks which have been the old limit. The old limit explains also another bug which you might have encountered or will encounter: It doesn't save the upgrades of 33+ stacks. To save them you need to manually move them higher than 32th position which can get annoying at times.

5.5.2.6 Experience Constants

Take note that two constants are getting declared in header_parties.py which influence the outcome of various scripts, the so called experience constants

```
player_loot_share = 10
hero_loot_share = 3
```

They are getting called in the script `calculate_main_party_shares`, `player_loot_share` additionally in the scripts `party_calculate_loot` and `party_give_xp_and_gold`, so they influence experience, gold and loot after a battle.

5.5.2.7 Fiefs' Association with each other

The game uses the slot `ßlot_village_bound_center"` to determine which fiefs are associated with each other. When you start a new game the script `"game_start"` does all the setup, which includes

[438] Conversation of Dalion and Earendil, Mount & Blade Modding Discord.

the distribution of villages to towns and castles. The Native rule is that each castle gets assigned one village, while towns can have several villages. It uses the distance to a center to choose who belongs to who. Search for the abovementioned slot on module_scripts.py to see the code.[439]

These are the Native rules, yours can be whatever you want them to be. The economy, recruiting, caravan transport, production of goods, prosperity of center, etc. are all defined on the Module System. If you want to remove/replace/do something else you will need to edit the respective code. It is not an easy “edit this line of code“, it is more like “edit this 100 lines of code spread on 50 different places and files“, so that means you need some level of experience with coding MaBL and testing your features first. Instead of using Native you can alternatively use an Open Source mod that already does that as your base.[440]

5.5.2.8 Other Notes

Note slot number, function of higher ones?[441]

Party and Lord AI (chapter about AI?)[442]

city assignment overwrites faction assignment at parties (remark)[443]

carries goods[444]

party initiative[445]

party flags[446]

[439] kalarhan, Modding Q&A.
[440] kalarhan, Modding Q&A.
[441] Lumos, Modding Q&A.
[442] Lav, Modding Q&A.
[443] Fire_and_Blood, Modding Q&A.
[444] Ikaguia, Modding Q&A.
[445] kalarhan, Modding Q&A.
[446] kalarhan, Modding Q&A.

5.5.3 Module Party Templates

```
Each party template record contains the following fields:
1) Party-template id: used for referencing party-templates in other files.
   The prefix pt_ is automatically added before each party-template id.
2) Party-template name.
3) Party flags. See header_parties.py for a list of available flags
4) Menu. ID of the menu to use when this party is met. The value 0 uses the default party encounter system.
5) Faction
6) Personality. See header_parties.py for an explanation of personality flags.
7) List of stacks. Each stack record is a tuple that contains the following fields:
  7.1) Troop-id.
  7.2) Minimum number of troops in the stack.
  7.3) Maximum number of troops in the stack.
  7.4) Member flags(optional). Use pmf_is_prisoner to note that this member is a prisoner.
  Note: There can be at most 6 stacks.
```

The file **module_party_templates.py** begins with the usual Python list: party_templates = [, followed by several templates that are hardwired into the game and should not be edited. You'll notice that the tuples in module_party_templates.py are very similar to those in module_parties.py, but the two are not interchangeable. Look for example at the following party template:

```
("village_farmers","Village Farmers",icon_peasant|pf_civilian,0,fac_innocents, merchant_personality,[(trp_farmer,5,10),(trp_peasant_woman,3,8)]),
```

It is a template which we all have encountered in-game. Parties of this template will be called "farmers", they are marked as civilians, they will behave cowardly in-game, and each of them has two troop stacks made up of farmers and peasant women. Breaking down the tuple fields yields:

```
1) Party-template id = "village_farmers"
2) Party-template name = "Village Farmers"
3) Party flags = icon_peasant, pf_civilian
4) Menu = 0
5) Faction = fac_innocents
6) Personality = merchant_personality
7) List of stacks:
  7.1) Troop-id = trp_farmer, trp_peasant_woman
  7.2) Minimum number of troops in the stack = 5 farmers, 3 peasant women
  7.3) Maximum number of troops in the stack = 10 farmers, 8 peasant women
  7.4) Member flags (optional) = None are set.
```

Keep in mind that there can be at most six different troop stacks in a party template (the same is valid for a party declared in module_parties.py).

Parties are not to be confused with party templates. In the simplest terms, party templates are a set of guidelines from which parties on the map are spawned. This marks the most notable difference between parties and party templates - parties are unique entities on the map, whereas templates do not physically exist in the game world. They serve only as a list of guidelines from which to spawn parties. Therefore, certain operations that use a party_id for input will not work when they are fed a party_template_id.

Parties that are spawned from a template do not have to be unique. There can be many parties of the same template; each will have a random number of troops depending on the player's level and the minimum or maximum troop limits defined in the template.

If you've followed the documentation since the beginning, you should be fairly adept at reading tuples by this point and you will have noticed the one field in this tuple that's unlike any other field we've encountered before: Field number 6, the Personality flags field.

5.5.3.1 Personality

As mentioned in the tuple breakdown, the Personality field determines party behaviour on the map. Here you can assign custom scores for Courage and Aggressiveness, or use one of the preset personalities such as merchant_personality. These presets are constants, each containing a Courage and an Aggressiveness score. The presets are all defined in header_parties.py, so you can open that file now and scroll to the bottom to see the constant definitions for yourself. You find them also listed hereinafter. There you will also notice the list of possible Courage and Aggressiveness settings.

The constant merchant_personality is used in many templates throughout the file. Parties with this personality will be friendly, they will not go out to attack the enemy or raid weaker parties. This is because merchant_personality sets the party's Aggressiveness to aggresiveness_0. A party with aggresiveness_0 will never attack another party, whereas normal combative parties with the preset soldier_personality will have aggresiveness_8. This will let them attack other parties if the attackers' faction is on bad terms with the defenders' faction, and if the would-be attackers aren't badly outnumbered.

Courage is the score that determines when parties will run away from another, larger party. Higher Courage means they will be less quick to turn away when the numbers aren't entirely favorable. merchant_personality parties have a Courage of 8, where soldier_personality have a Courage of 11.

These settings scale from 0 to 15, allowing you to precisely set the desired behavior for your party templates. New modders, however, are recommended to stick with the presets. They cover the full range of personalities you should need for your first mod.

Finally, for bandit templates, there is the flag banditness. This causes the bandit party to constantly consider other nearby parties as prey, and if the prey is carrying significant amounts of gold and/or trade goods, the bandit party will attack. Ideally, a bandit party should have low aggressiveness or low troop numbers so that it does not attack soldier parties. NOTE: You may not see the banditness flag directly used at party templates but it is listed in headers_parties.py as part of bandit_personality.

The exact functionality of aggressiveness/courage scores (as well as things like 'banditness') on whether parties will approach one another or run away is hard-coded and can therefore not be altered, as is the pathfinding of AI parties.[447]

List of available settings:

courage_x The lowest courage level x for a party is 4, extremely cowardly, it will run away from much smaller enemy parties. A level of 8 signifies a party with neutral courage. A party with a courage level of 15 is extremly brace and will be willing to fight much larger enemy parties.

aggressiveness_x The lowest aggressiveness level x for a party is 0, unaggressive, it will never actively attack another party. On the contrary, a party with an aggressiveness level of 15 is extremely aggressive.

banditness

Following predefined personality constants are given, you are free to add your own behind them.

```
soldier_personality = aggressiveness_8 | courage_9
merchant_personality = aggressiveness_0 | courage_7
escorted_merchant_personality = aggressiveness_0 | courage_11
bandit_personality    = aggressiveness_3 | courage_8 | banditness
```

5.5.3.2 Creating a New Party Template

You can of course create new party templates. For easiest insurance that they will be used correctly, define a new `slot_faction_reinforcements_YOURLETTER` slot in module_constants.py. Populate it then in the script “initialize_faction_troop_types“ with the others and call it in the appropriate reinforcement scripts. A quick search for slot_faction_reinforcements_a in module_scripts.py will show you the places you need to edit. There are not many, and other than declaring the slot in constants and creating the new template(s), that is the only file that needs editing by you.[448]

[447] Caba‘drin, Modding Q&A.

[448] Caba‘drin, Modding Q&A.

5.5.4 Ground Specs

map textures and the scene textures

world terrain stuff?[449]

ground specs, green tint on large sceneprops (not sure yet where to put that)[450]

[449] Yoshiboy, Data folder modular.

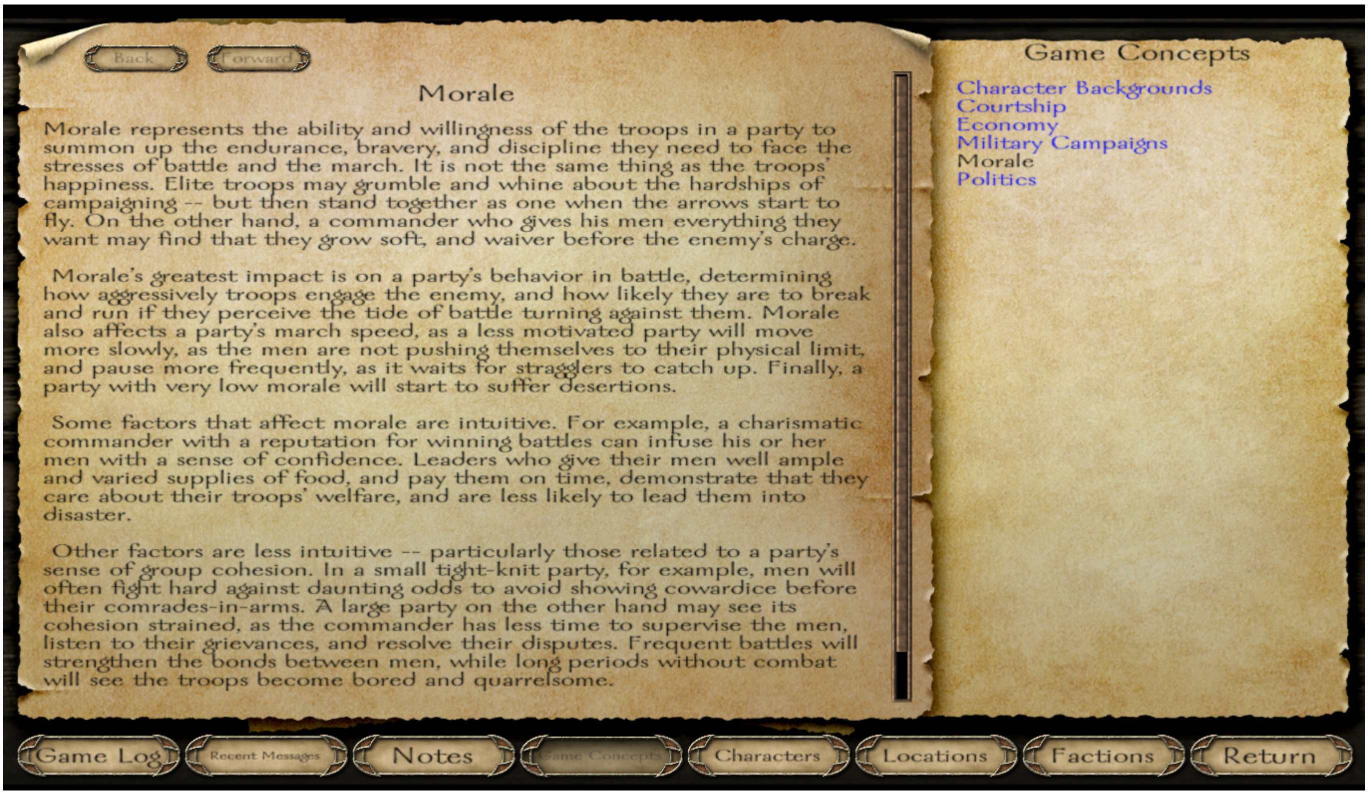
[450] Docm30, Modding Q&A, and Lumos, Modding Q&A.

5.6 UI-related Module System Files

5.6.1 Module Info Pages

```
Each quest record contains the following fields:
1) Info page id: used for referencing info pages in other files. The prefix ip_ is
     automatically added before each info page id.
2) Info page name: Name displayed in the info page screen.
```

module_info_pages contains informations about morale, economy, courtship, politics, character backgrounds and military campaigns which the player can open up at the World Map by clicking “Notes“ and then “Game Concepts“. The character limit for entries is $2^\wedge 14 = 16384$. Forward slashes \ don’t count as characters because they are not displayed in game, they serve only for easier editing purposes in the .py file, ^ characters do count though.[451]



Morale

Morale represents the ability and willingness of the troops in a party to summon up the endurance, bravery, and discipline they need to face the stresses of battle and the march. It is not the same thing as the troops' happiness. Elite troops may grumble and whine about the hardships of campaigning -- but then stand together as one when the arrows start to fly. On the other hand, a commander who gives his men everything they want may find that they grow soft, and waiver before the enemy's charge.

Morale's greatest impact is on a party's behavior in battle, determining how aggressively troops engage the enemy, and how likely they are to break and run if they perceive the tide of battle turning against them. Morale also affects a party's march speed, as a less motivated party will move more slowly, as the men are not pushing themselves to their physical limit, and pause more frequently, as it waits for stragglers to catch up. Finally, a party with very low morale will start to suffer desertions.

Some factors that affect morale are intuitive. For example, a charismatic commander with a reputation for winning battles can infuse his or her men with a sense of confidence. Leaders who give their men well ample and varied supplies of food, and pay them on time, demonstrate that they care about their troops' welfare, and are less likely to lead them into disaster.

Other factors are less intuitive -- particularly those related to a party's sense of group cohesion. In a small tight-knit party, for example, men will often fight hard against daunting odds to avoid showing cowardice before their comrades-in-arms. A large party on the other hand may see its cohesion strained, as the commander has less time to supervise the men, listen to their grievances, and resolve their disputes. Frequent battles will strengthen the bonds between men, while long periods without combat will see the troops become bored and quarrelsome.

[451] Dalion, Mount & Blade Modding Discord.

5.6.2 Module Dialogs

```
During a dialog, the dialog lines are scanned from top to bottom.
If the dialog-line is spoken by the player, all the matching lines are displayed for the player to pick from.
If the dialog-line is spoken by another, the first (top-most) matching line is selected.

Each dialog line contains the following fields:
1) Dialogue partner: This should match the person player is talking to.
   Usually this is a troop-id.
   You can also use a party-template-id by appending '|party_tpl' to this field.
   Use the constant 'anyone' if you'd like the line to match anybody.
   Appending '|plyr' to this field means that the actual line is spoken by the player
   Appending '|other(troop_id)' means that this line is spoken by a third person on the scene.
      (You must make sure that this third person is present on the scene)
2) Starting dialog-state:
   During a dialog there's always an active Dialog-state.
   A dialog-line's starting dialog state must be the same as the active dialog state, for the line to be a possible candidate.
   If the dialog is started by meeting a party on the map, initially, the active dialog state is "start"
   If the dialog is started by speaking to an NPC in a town, initially, the active dialog state is "start"
   If the dialog is started by helping a party defeat another party, initially, the active dialog state is "party_relieved"
   If the dialog is started by liberating a prisoner, initially, the active dialog state is "prisoner_liberated"
   If the dialog is started by defeating a party led by a hero, initially, the active dialog state is "enemy_defeated"
   If the dialog is started by a trigger, initially, the active dialog state is "event_triggered"
3) Conditions block (list): This must be a valid operation block. See header_operations.py for reference.
4) Dialog Text (string):
5) Ending dialog-state:
   If a dialog line is picked, the active dialog-state will become the picked line's ending dialog-state.
6) Consequences block (list): This must be a valid operation block. See header_operations.py for reference.
```

```
7) Voice-over (string): sound filename for the voice over. Leave here empty for no voice over
```

In this Subchapter we will be examining **module_dialogs.py**, by far the largest file in the module system and one of the most important, as it contains all dialogue in Mount&Blade. Any new dialogue that you wish to create will go into this file. Understand that this file is read from top to bottom each time a dialog is started. When the criteria for a tuple is met, then it will be executed.

The module_dialog.py file is a bit more complex than it used to be. Looking at the first few tuples, you will see that the target “anyone“ is used with the various start-dialog states (explained later). There conditions are getting checked and registries, string registries and global variables set. What may seem odd is that each one's dialog text states that the dialog is not displayed. These dialogs set up these pieces of information to be used further down in the file. This way, these code blocks only need to be typed in once, not in every conversation.

If the dialog partner does not fit the criteria for that tuple, then it is skipped. This makes it easier for you to add new generic NPCs, village elders, etc. without having to worry about setting their conversations. Take a quick look at one of them near the top:

```
dialogs = [
  [anyone ,"start", [(store_conversation_troop, "$g_talk_troop"),
                     (store_conversation_agent, "$g_talk_agent"),
                     (store_troop_faction, "$g_talk_troop_faction", "$g_talk_troop"),
#                     (troop_get_slot, "$g_talk_troop_relation", "$g_talk_troop", slot_troop_player_relation),
                     (call_script, "script_troop_get_player_relation", "$g_talk_troop"),
                     (assign, "$g_talk_troop_relation", reg0),
                     ...
                     (try_begin),
                       (this_or_next|is_between, "$g_talk_troop", village_elders_begin, village_elders_end),
                       (is_between, "$g_talk_troop", mayors_begin, mayors_end),
                       (party_get_slot, "$g_talk_troop_relation", "$current_town", slot_center_player_relation),
                     (try_end),
```

There is more, but this will give you the idea. This will be checked when you **start** a conversation with **anyone**. Looking at the try block, we see that it checks if "$g_talk_troop" is listed between the constants "village_elders_begin" and "village_enders_end". Since the operation `this_or_next` is used, this condition and the next condition are read as an OR condition. This would mean that if the first condition is not met but the second condition (is_between, "$g_talk_troop", mayors_begin, mayors_end) is true, the rest of the code block is still executed.

The used constants are defined in module_constants.py. This is why, if you want to add a village elder, it should be placed between `"trp_village_1_elder"` and `"trp_merchants_end"` as defined in that file to take advantage of the settings pre-defined for village elders.

Let's scroll down a bit, and find a more specific dialog, namely the start conversation with Ramun the slave trader:

```
 [trp_ramun_the_slave_trader, "start", [
  (troop_slot_eq, "$g_talk_troop", slot_troop_met_previously, 0),
  ], "Good day to you, {young man/lassie}.", "ramun_introduce_1",[]],
```

Breaking down the tuple field of Ramun's opening statement:

```
1)Dialogue partner : trp_ramun_the_slave_trader
2)Starting Dialog-State : "start"
3)Conditions block : (troop_slot_eq, "$g_talk_troop", slot_troop_met_previously,
   0),
4) Dialog text string : "Good day to you, {young man/lassie}."
5) Ending dialog-state : "ramun_introduce_1"
6) Consequences : []
```

The most important things to note in this tuple are the dialog-states, both the starting and ending dialog-states. We will delve into these more deeply now.

The ending dialog-state `"ramun_introduce_1"` is what leads a conversation from one line to the next. The ending dialog-state can be anything you like but there must be another tuple with a matching starting dialog-state. For example, if we were to make a tuple with the ending state `"blue_oyster"`, this would lead into any tuple with the starting state `"blue_oyster"`. There must be an exact match; if no match exists, build_module.bat will throw an error upon trying to build.

If there are multiple tuples with the starting state `"blue_oyster"`, something special happens. If the tuples are spoken by the player, they result in a menu where the player can choose between the provided tuples. In Vanilla M &B the game only has room for five dialogue options at a time while the limit is 1024 at the Warband engine. The lines can't be too long, either, or they will either shrink or spill out of their box. If it is the player's answer, it will shrink as much as it can before getting out. If it is the other partner's dialog line, it will spill out right away[452]. If the tuples are spoken by an NPC, the module system will use the first tuple in module_dialogs.py for which all conditions are met, even if there are multiple lines that qualify.

[452] Dalion, Mount & Blade Modding Discord.

To end a conversation, you must use the ending dialog-state `"close_window"`. To start a conversation, there are several special starting dialog-states from which you can choose. We will refer to these as initial dialog-states . Here is the full list of initial dialog-states:

start is considered when speaking to a NPC in a scene or when a conversation is triggered inside a scene.

party_encounter is considered when encountering another party on the overland map or in a scene.

party_relieved is considered when assisting a battling party on the overland map, once the player has won the fight.

prisoner_liberated is considered when the player defeats an enemy party with one or more Hero prisoners.

enemy_defeated is considered when the player defeats an enemy party led by a Hero.

event_triggered is considered when a dialogue is triggered by an operation while not in a scene.

As you can see, each initial dialog-state is designed for a specific situation. It will not be considered anywhere outside its situation.

5.6.2.1 Requirements and the Condition Block

The dialogue interface is very flexible, and can be used in a great number of ways. It handles both events on the overland map and events in scenes. It allows dialogues to be triggered whenever you want them. In the following segments, we examine how to use the dialogue interface to its fullest.

As we have outlined in the last segment, a dialogue line will only be considered if all its requirements are met. First of all, the player must be speaking to the correct troop. A line with `trp_ramun_the_slave_trader` will not be considered if the player is addressing `trp_constable_hareck`. The constant **anyone** may be used if the line is to be spoken by anyone the player is addressing at the time.

Second, the starting dialog-state must be in accordance with the situation - either the dialogue line is initiated by an initial dialog-state, or the line follows from a matching ending dialog-state in another line. If the starting dialog-state does not meet specifications, it won't be considered for use.

Thirdly, the same logic of the previous two points applies to the *conditions block*. Unless all of a line's conditions are met, the line will not be considered. If either the conditions or the starting dialog-state are erroneous, you will experience problems; either build_module.bat will throw an error, or the erroneous dialogue lines will simply freeze in-game, since their ending dialog-states will not be able to find another tuple to activate. This is the reason why you must be careful with conditions blocks. Make sure you don't break your own dialogue by planting conditions which are not properly set.

However, when everything is working in harmony, conditions blocks can be very powerful since they can contain try blocks. You can call slots from inside a conditions block, and then use the result in a condition operation inside the same block. You can set registers and string registers in order to use them in the actual dialogue.

You can observe the use of a conditions block in the tuple of module_dialogs.py which is used to talk with Ramun the slave trader:

```
[trp_ramun_the_slave_trader, "start", [(troop_slot_eq, "$g_talk_troop",
    slot_troop_met_previously, 0),],
```

This block contains only one condition, which requires the variable `slot_troop_met_previously` of "$g_talk_troop" to be equal to 0. This is because all registers and variables are equal to 0 at the beginning of a new game. And if you look at the next tuple, you will notice the consequences block:

```
[trp_ramun_the_slave_trader|plyr, "ramun_introduce_1", [], "Forgive me, you look
    like a trader, but I see none of your merchandise.", "ramun_introduce_2",[
  (troop_set_slot, "$g_talk_troop", slot_troop_met_previously, 1),
]],
```

The `troop_set_slot` operation in this block sets the variable `slot_troop_met_previously` of "$g_talk_troop" (to whom you are talking) to 1 after the line has been displayed. In other words, after this line has been displayed once, it will never be displayed again - because afterwards, the variable `slot_troop_met_previously` will no longer be equal to 0. The dialogue system will then ignore this line and instead go to the next tuple in the file which meets requirements:

```
[trp_ramun_the_slave_trader,"start", [], "Hello, {playername}.", "ramun_talk",[]],
```

The only requirements on this line are that the player must be speaking to Ramun in a scene. It will always be selected when speaking to Ramun after he has delivered his speech.

It is important to know that various variables are set in the first three tuples of this file. They are set when you talk with **anyone**. Each has its specific dialog start state, but only covers conversations between you and someone you are specifically targeting for a conversation. "$g_talk_troop" is just one of the variables cataloged and set in this process. In the first few lines of the first tuple, they also store your relation with "$g_talk_troop" from reg0, which is calculated in the script `troop_get_player_relation` (check it out in module_scripts.py) and stored in "$g_talk_troop_relation":

```
[anyone ,"start", [(store_conversation_troop, "$g_talk_troop"),
                    (store_conversation_agent, "$g_talk_agent"),
                    (store_troop_faction, "$g_talk_troop_faction", "$g_talk_troop")
                      ,
#                    (troop_get_slot, "$g_talk_troop_relation", "$g_talk_troop",
   slot_troop_player_relation),
                    (call_script, "script_troop_get_player_relation", "
                      $g_talk_troop"),
                    (assign, "$g_talk_troop_relation", reg0),
```

Things are done this way so you won't have to do this kind of thing every time you create a new dialog block. If you will be having special dialogs that will take into consideration things like your relation with the target, then it would be good to scan through these (or the specific start-dialog state) and see if what you will be checking is already set.

Pages 27-30

5.6.2.2 Declarations for Dialogs[453]

anyone dialog will start with anyone if you send the game to its dialog state. I.e. it won't check for the specific troop id.

plyr dialog answer will be played from the player agent.

party_tpl the dialog will be issued only when you meet a party of specified template on global map. See Native examples for the syntax.

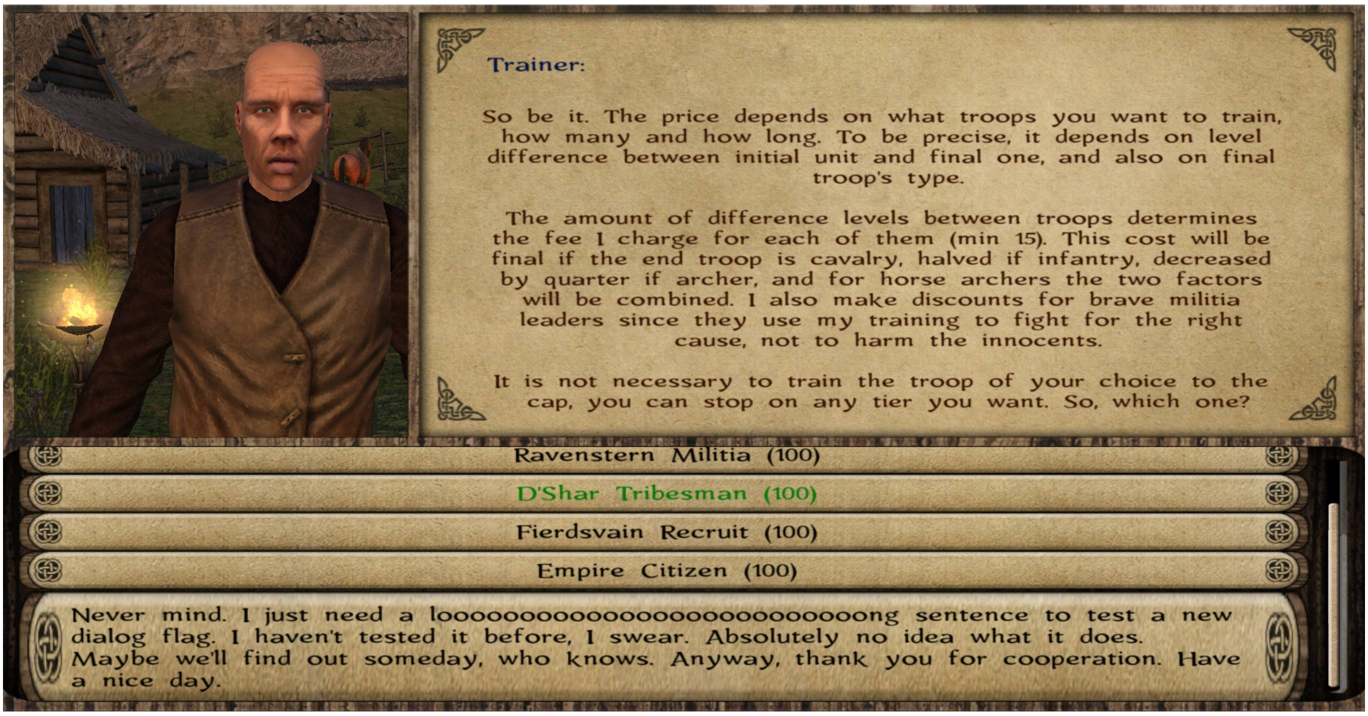
auto_proceed dialog answer won't require player to press left mouse button to continue further as dialogs usually do. It is instant, so the player can't even see the text. It is assumed that this flag is for assigning some important variables that will be used in the dialog it jumps to.

multi_line is for **plyr** answers only. It makes them behave like your partner's answers i.e. lets the text jump to the next line instead of shrinking it as it would be otherwise as can be seen in the picture below.

repeat_for_factions creates a list of factions as player answers, the amount of options will depend on how many factions module_factions.py file has.

repeat_for_parties creates a list of parties as player answers, the amount of options will depend on how many parties module_parties.py file has.

[453] Dalion



Trainer:

So be it. The price depends on what troops you want to train, how many and how long. To be precise, it depends on level difference between initial unit and final one, and also on final troop's type.

The amount of difference levels between troops determines the fee I charge for each of them (min 15). This cost will be final if the end troop is cavalry, halved if infantry, decreased by quarter if archer, and for horse archers the two factors will be combined. I also make discounts for brave militia leaders since they use my training to fight for the right cause, not to harm the innocents.

It is not necessary to train the troop of your choice to the cap, you can stop on any tier you want. So, which one?

Ravenstern Militia (100)

D'Shar Tribesman (100)

Fierdsvain Recruit (100)

Empire Citizen (100)

Never mind. I just need a looooooooooooooooooooooooooong sentence to test a new dialog flag. I haven't tested it before, I swear. Absolutely no idea what it does. Maybe we'll find out someday, who knows. Anyway, thank you for cooperation. Have a nice day.

repeat_for_troops creates a list of troops as player answers, the amount of options will depend on how many troops module_troops.py file has.

repeat_for_100 restricts the list of answer options to 100.

repeat_for_1000 restricts the list of answer options to 1000.

All the repeat_for_x declarations create a list of player options (so `plyr` is required), where each option is a separate object. The amount of options depends on the last number in the respective ID_xxx.py file (max 1024). The declarations should be used on pair with the operation `store_repeat_object` (in both brackets of the dialog answer, so in the condition and consequences ones). The example is shown here: https://imgur.com/a/dizGEM9 (need to integrate that in some way).

5.6.2.3 Dialog Notes[454]

The first section

```
"[ ...some code here... ]"
```

is a block that exists to test a condition or series of conditions. It's only purpose is to return TRUE or FALSE. And every single dialog gets checked until a TRUE result occurs.

Secondly, the parts after "start" can only be accessed IF "start" happened, their branch is active AND their own condition block is TRUE. So, if you want something that always happens when

[454] Taken from *An Introduction to Module Syntax and Usage* by xenoargh

Generalissimo EvilGuy meets your player on a Tuesday, but you want him to do something else on Wednesday, his "start" might look like this:

```
#Start of Dialog block
[
#See header_dialogs for what can be here besides "anyone".  All Dialogs must have a "start".
anyone, "start",
#Start condition block here
[
(eq, "$g_talk_troop", "trp_generalissimo_evilguy"),#ONLY RUN ME IF EVILGUY
(try_begin),
  (eq, "$g_day_of_week", 2),#DO THIS TUESDAYS
  (assign, "$g_conversation_temp",1),
  (str_store_string, s17, "@This is my Tuesday conversation start"),
(else_try),
  (eq, "$g_day_of_week", 3),#WEDNESDAYS
  (assign, "$g_conversation_temp",2),
  (str_store_string, s17, "@This is my Wednesday conversation start"),
(else_try),
  (assign, "$g_conversation_temp",3),#IF ALL ELSE FAILS...
  (str_store_string, s17, "@This is what I say any other day of the week"),
(try_end),
],
#Condition is TRUE; we're talking to EvilGuy, now we need our custom text:
"s17",
#Now we the Player to be able to say something, even if it's something prosaic like, "leave", to exit this Dialog.
"player_response_or_responses_here",
#Consequences, if any:
[],
#End of dialog block
],
```

I don't like doing Dialogues using a Troop in that first block; it makes it easier to accidentally create logic errors. I prefer to do it in the Conditions block, like so:

```
[anyone,"start", [(eq, "$g_talk_troop", "trp_player_castellan"),], "What can I do for you {playername}", "castellan_talk",[]],  [/code]
That means that I can have multiple starts for that Troop, depending on the conditions block returning TRUE, like this:
[code]
[anyone,"start",
[
(eq, "$g_talk_troop", "trp_player_castellan"),
(eq, "$g_some_game_global", 1),
], "What can I do for you {playername}", "castellan_talk",[]],
[anyone,"start",
[
(eq, "$g_talk_troop", "trp_player_castellan"),
(eq, "$g_some_game_global", 2),
], "Oh, it's {playername}, that scummy devil, come to bother me again!", "castellan_talk",[]],
[/code]
The only major exception to this is dealing with Party encounters where you may be talking to any random guy in the Party, for example, the Looter's start:
[code][party_tpl|pt_looters|auto_proceed,"start", [(eq,"$talk_context", tc_party_encounter),(encountered_party_is_attacker),], "{!}Warning: This line should never be displayed.", "looters_1",[
    (str_store_string, s11, "@It's your money or your life, {mate/girlie}. No sudden moves or we'll run you through."),
    (str_store_string, s12, "@Lucky for you, you caught me in a good mood. Give us all your coin and I might just let you live."),
    (str_store_string, s13, "@This a robbery, eh? I givin' you one chance to hand over everythin' you got, or me and my mates'll kill you. Understand?"),
    (store_random_in_range, ":random", 11, 14),
    (str_store_string_reg, s4, ":random"),
    (play_sound, "snd_encounter_looters")
  ]],[/code]
Note there that there are not two but four things that must be TRUE:
1. It's a Party encounter on the overworld.
2. The Party Template is pt_looters.
3. $talk_context == tc_party_encounter.
4. encountered_party_is_attacker must be TRUE
```

5.6.2.4 Conversation with Multiple People

Link

5.6.2.5 Other notes

Dialog states[455]

[455] cwr (credit), Modding Q&A.

Rare dialog setup[456]

party template dialog[457]

[456] jacobhinds, Modding Q&A, and Somebody, Modding Q&A.

[457] cwr (credit), Modding Q&A.

5.6.3 Module Quests

```
Each quest record contains the following fields:
1) Quest id: used for referencing quests in other files. The prefix qst_ is automatically added before each quest-id.
2) Quest Name: Name displayed in the quest screen.
3) Quest flags. See header_quests.py for a list of available flags
4) Quest Description: Description displayed in the quest screen.
Note that you may call the opcode setup_quest_text for setting up the name and description
```

module_quests.py is a small, simple file. It contains quests, including all the text related to those quests. Putting new quests here allows them to be activated via the module system, so that operations can read the quest's current status and use that status as a condition operation. It also defines a quest object, for which the aforementioned slots can be used. Example of a quest:

```
 ("deliver_message", "Deliver Message to {s13}", qf_random_quest,
  "{!}{s9} asked you to take a message to {s13}. {s13} was at {s4} when you were given this quest."
  ),
```

This quest is one of many generic delivery request a king may give you.[458] Tuple breakdown:

```
1) Quest id = "deliver_message"
2) Quest Name = "Deliver Message"
3) Quest flags = qf_random_quest
4) Quest Description = "{s9} asked you to take a message to {s13}. {s13} was at {s4} when you were given this quest."
```

This quest uses string registers (s9, s13, and s4) to indicate who gave the quest(s9), the recipient (s13), and where the target was last seen (s4). Looking at similar quests, it would seem that s9 is always used as the requester's name, s13 is the intended target troop/party and s4 is used as their last location.

In the progress of creating a new quest, you will be using some of the in Subchapter 5.2.1 mentioned slots for your quest. This way, the value can be set in one place, and if it needs to be changed you only need to change it in one place. The other benefit is that if the player has more than one quest active, you don't have to worry that you might be over-writing variables we defined. Each quest (like all tuple objects) has its own slots.

You will quickly notice that it takes only a few changes to make a new quest, but incorporating it

[458] This specific quest is defined as the first governer quest in module_constants.py, as you can read in the comment above it. Again, if comments are given, read them. In most cases they contain useful informations for you.

into the game requires a lot more work. You need a place for the player to get the quest, the “actors“ that will play a part in the quest and some dialog to make it interesting.

5.6.3.1 Bare-bones for New Quests[459]

Here is the bare-bones way to create new quests with MABL. First, you need a consequence block (at the end of a dialogue line, for example) that begins your quest, such as this one:

```
[[assign,"$do_stuff_quest_active",1],
[setup_quest_text,"qst_do_stuff"],
[start_quest,"qst_do_stuff"]]],
```

[[assign,"$do_stuff_quest_active",1], is just a string, not related to the actual quest but handy for coding quest-related condition and consequence blocks later. [setup_quest_text,"qst_do_stuff"], activates the quest text for qst_do_stuff. From experience, this is not strictly necessary but it pays to play things safe. [start_quest,"qst_do_stuff"]]], is the consequence that will actually begin your quest. You can use this kind of consequence block in module_dialogs.py, module_game_menus.py and module_triggers.py (and in others probably as well).

Next, you have to input your quest text at the bottom of module_quests.py.

```
("do_stuff", "Do some really amazing stuff", qf_show_progression,
"Some guy asked you to do some really amazing stuff, and he'll give you some gold
    ."
),
```

("do_stuff", "Do some really amazing stuff", qf_show_progression, is where most of the action happens. The first string sets up the name of your quest, the second string sets up the short text as you will see it in the top left-hand bar of the quest window, and then come the quest flags, which determine whether you want this to be a random quest (qf_random_quest) OR if you want to show what percentage of this quest is completed (qf_show_progression). It should be possible to put a 0 there if you don’t want any quest flags. "Some guy asked you to do some really amazing stuff, and he’ll give you some gold ." is the long quest text that will show up in the large top-right window of the quest screen. And), closes your block as you might have guessed already. Having done this, input whatever content (dialogue, fights and so on) you want this quest to have.

With everything else done the time has come to close your quest.

```
[[add_xp_as_reward,100],
[complete_quest,"do_stuff"],
[set_quest_progression,"qst_do_some_more_stuff",40],
```

[459] Winter, Modding Q&A, might be outdated though. Thus verification needed.

```
[assign,"$quest_succeeded_do stuff",1]]],
```

[[add_xp_as_reward,100], is a way to add XP as a reward for your quest. There are other ways to reward the player as well, such as add_gold_to_troop, add_item_to_troop, and so on. [complete_quest ,"do_stuff"], completes your quest. You can also use succeed_quest or fail_quest, but they are not really necessary when you can use a simple string to do the same job. If you have set the qf_show_progression flag on your quest, use the operation [set_quest_progression,"qst_do_some_more_stuff ",40], to adjust the percentage shown. 40, in this example, will show the quest "do_some_more_stuff" as 40 % complete. [assign,"$quest_succeeded_do stuff",1]]], is a string you can use for future conditions blocks which depend on whether the player has succeeded, failed or otherwise finished the quest. You can make the string anything you like and use any number of strings for this purpose.

This is all you need for a basic quest. More complex ones will require more complicated finagling, especially quests involving heroes.

5.6.3.2 Quest Flags

If you wish to make a quest without any flags, simply put a 0 in the flags field. Otherwise you have the following two available:

qf_show_progression shows what percentage of this quest is completed.

qf_random_quest determines that this quest is a random one.

quests and slots (also later messages at that page)[460]
quests[461]

[460] Khamukkamu and kalarhan, Modding Q&A.
[461] kalarhan, Modding Q&A.

5.6.4 Module Meshes

```
Each mesh record contains the following fields:
 1) Mesh id: used for referencing meshes in other files. The prefix mesh_ is automatically added before each mesh id.
 2) Mesh flags. See header_meshes.py for a list of available flags
 3) Mesh resource name: Resource name of the mesh
 4) Mesh translation on x axis: Will be done automatically when the mesh is loaded
 5) Mesh translation on y axis: Will be done automatically when the mesh is loaded
 6) Mesh translation on z axis: Will be done automatically when the mesh is loaded
 7) Mesh rotation angle over x axis: Will be done automatically when the mesh is loaded
 8) Mesh rotation angle over y axis: Will be done automatically when the mesh is loaded
 9) Mesh rotation angle over z axis: Will be done automatically when the mesh is loaded
10) Mesh x scale: Will be done automatically when the mesh is loaded
11) Mesh y scale: Will be done automatically when the mesh is loaded
12) Mesh z scale: Will be done automatically when the mesh is loaded
```

Some intro text here

5.6.4.1 Mesh Flags

render_order_plus_1

5.6.5 Module Game Menus

```
Each game menu is a tuple that contains the following fields:
1) Game-menu id (string): used for referencing game-menus in other files.
   The prefix menu_ is automatically added before each game-menu-id
2) Game-menu flags (int). See header_game_menus.py for a list of available flags.
   You can also specify menu text color here, with the menu_text_color macro
3) Game-menu text (string).
4) mesh-name (string). Not currently used. Must be the string "none"
5) Operations block (list). A list of operations. See header_operations.py for reference.
   The operations block is executed when the game menu is activated.
6) List of Menu options (List).
   Each menu-option record is a tuple containing the following fields:
6.1) Menu-option-id (string) used for referencing game-menus in other files.
     The prefix mno_ is automatically added before each menu-option.
6.2) Conditions block (list). This must be a valid operation block. See header_operations.py for reference.
     The conditions are executed for each menu option to decide whether the option will be shown to the player or not.
6.3) Menu-option text (string).
6.4) Consequences block (list). This must be a valid operation block. See header_operations.py for reference.
     The consequences are executed for the menu option that has been selected by the player.
Note: The first Menu is the initial character creation menu.
General structure: (menu-id, menu-flags, menu_text, mesh-name, [<operations>], [<options>]),
```

Pages 21-23, some intro text here

5.6.5.1 Menu Flags

mnf_auto_enter lets the player automatically enter the town with the first menu option.

mnf_disable_all_keys disables the usage of the keys P, I, C.

mnf_enable_hot_keys enables the usage of the keys P, I, C.

mnf_join_battle lets the game consider this menu when the player joins a battle.

mnf_scale_picture scales the menu picture to offest screen aspect ratio.

menu text colour[462]

Passages in scenes[463]

only things executed between menu condition and consequences[464]

Passages[465]

enable hotkeys[466]

hard-coded alphabetic sorting in note list[467]

menus are hard-coded presentations[468]

Perhaps useful but probably outdated index of menus[469]

[462] Caba‘drin, Modding Q&A.
[463] Somebody, Modding Q&A.
[464] cmpxchg8b, Modding Q&A.
[465] Somebody, Modding Q&A.
[466] Somebody, Modding Q&A.
[467] The_dragon, Modding Q&A.
[468] kalarhan, Modding Q&A.
[469] Khalid ibn Walid (credit), 0.89x game_menus - index.

5.6.6 Module Presentations

```
Each presentation record contains the following fields:
1) Presentation id: used for referencing presentations in other files. The prefix prsnt_ is automatically added before each presentation id.
2) Presentation flags. See header_presentations.py for a list of available flags
3) Presentation background mesh: See module_meshes.py for a list of available background meshes
4) Triggers: Simple triggers that are associated with the presentation
```

Some intro text here

5.6.6.1 Presentation Flags

prsntf_manual_end_only

prsntf_read_only

5.6.6.2 Text Flags

tf_center_justify

tf_double_space

tf_left_align

tf_right_align

tf_scrollable

tf_scrollable_style_2

tf_single_line

tf_vertical_align_center

tf_with_outline

Only one active presentation at any time.[470]

New menu background. [471]

prsntf_read_only flag.[472]

[470] Somebody, Modding Q&A.

[471] Somebody, Modding Q&A.

[472] Caba‘drin, Modding Q&A.

overlay_set_area_size.[473]

Arena training presentation[474]

Something about a flag and dxt7[475]

presentation coordinates [476]

mouse hovers above troop name[477]

Rendering order[478]

tf_right_align[479]

text flag[480]

presentation basics[481]

prsnt_game_escape[482]

presentation get text box s[483]

0 valid overlay id[484]

overlay scales up[485]

worldmap presentations and stuff[486]

reference for designing presentations[487]

no background for presentation[488]

presentation triggers[489]

check if presentation is finished[490]

info bits about presentations[491]

Kuba helper presentation[492]

Presentation reference examples[493]

[473] Somebody, Modding Q&A.
[474] Somebody, Modding Q&A.
[475] Vornne, Modding Q&A.
[476] Vornne, Modding Q&A.
[477] Caba‘drin, Modding Q&A.
[478] Vornne and Lav, Modding Q&A.
[479] cmpxchg8b, Modding Q&A.
[480] Caba‘drin, Modding Q&A.
[481] Vornne and MadVader, Modding Q&A.
[482] The_dragon, Modding Q&A, and Lumos, Modding Q&A.
[483] Somebody, Modding Q&A, Modding Q&A and Modding Q&A.
[484] MadocComadrin, Modding Q&A.
[485] Lav, Modding Q&A.
[486] Ikaguia and Lav, Modding Q&A.
[487] DerGreif (credit), Modding Q&A.
[488] kalarhan, Modding Q&A.
[489] kalarhan, Modding Q&A.
[490] kalarhan, Modding Q&A.
[491] kalarhan, Modding Q&A.
[492] kalarhan, Modding Q&A.
[493] kalarhan, Modding Q&A.

presentations stuff[494]

Avoid using reg0 at presentations[495]

set background mesh[496]

[494] Arch3r (credit), Modding Q&A.

[495] kalarhan, Modding Q&A.

[496] Somebody and kalarhan, Modding Q&A.

5.7 Audio-related Module System Files

5.7.1 Module Music[497]

```
Each track record contains the following fields:
1) Track id: used for referencing tracks.
2) Track file: filename of the track.
3) Track flags: see header_music.py for a list of available flags.
4) Continue Track flags: Shows in which situations or cultures the track can
    continue playing. See header_music.py for a list of available flags.
```

module_music.py is where the music tracks of the module are getting recorded and defined. The file simply consists of a list of tuples. Observe:

```
("ambushed_by_vaegir", "ambushed_by_vaegir.ogg", mtf_culture_2|mtf_sit_ambushed|
    mtf_sit_siege, mtf_sit_fight|mtf_sit_multiplayer_fight|mtf_culture_all),
```

This is a standard music track called `"ambushed by vaegir"` which gets played when the player gets ambushed under certain circumstances. Lets do a short tuple examination:

```
"ambushed_by_vaegir"
```

The name of the track as known by the code. You can reference to it at other places in the Module System with `track_ambushed_by_vaegir`.

```
"ambushed_by_vaegir.ogg"
```

The name of the physical music file. If `mtf_module_track` is present in the third field, the file is in (module direction)/Music. Otherwise it is a Native file in (M&B direction)/Music. The first mistake which you will probably make is to forget to put the `mtf_module_track` flag when adding your own track.

```
mtf_culture_2|mtf_sit_ambushed|mtf_sit_siege
```

The play conditions, so when the track will play: in what situation and in what culture? In our example here: The track will play only if the culture is 2 (Vaegir) AND the situation is either Ambush OR Siege. Not having culture flags is the same as putting `mtf_culture_all` (will match all cultures) - this may be confusing, but this is what M&B does.

```
mtf_sit_fight|mtf_sit_multiplayer_fight|mtf_culture_all),
```

The continue condition, so whether the track continues to play when the situation and/or culture changes (otherwise it will fade out). In our example here: The track will continue to play in any

[497] Most parts of the explanations in this chapter are taken from Music modding reference by MadVader.

culture AND if the situation is Ambush OR Siege OR Fight. The track will also continue to play at the multiplayer. Note that the start play conditions are automatically added to continue play conditions (and it makes sense to do so - see process_music.py if you are curious). Usually you will ignore it (stick with the culture in the play condition) or put `mtf_culture_all` (when you want maximum chance of continuing play - if your play field has no `mtf_culture_x`, this is implied and can be omitted).

5.7.1.1 Music Track Flags

The following music track flags are available for use (describing their Native behaviour):

mtf_culture_1 lets the music play if the culture is culture 1 (Swadia).

mtf_culture_2 lets the music play if the culture is culture 2 (Vaegirs).

mtf_culture_3 lets the music play if the culture is culture 3 (Khergits).

mtf_culture_4 lets the music play if the culture is culture 4 (Nords).

mtf_culture_5 lets the music play if the culture is culture 5 (Rhodoks).

mtf_culture_6 lets the music play if the culture is culture 6 (Sarranids).

mtf_culture_all will allow the music to continue playing at any culture which handles the case when the player is hopping across the invisible border regularly.[498]

mtf_looping loops the track until continue conditions are not met (anymore).

mtf_module_track must be added to custom tracks for the mod located in the music folder at the module directory.

mtf_persist_until_finished plays the track until finished and nothing can stop it. Use with care!

mtf_sit_ambient_music sets the music preference to have an ambient setting. This flag has been introduced with VC and needs first to be added to the header_music.py if using the Native MS (mtf_sit_ambient_music = 0x02000000). A simple plugin and use of this flag is not possible, look up its usage at the VC MS.

mtf_sit_ambushed sets the music situation when an ambush is happening or if the odds against the player in a battle are worse than 2:1.

[498] MadVader, Modding Q&A.

mtf_sit_arena sets the music situation when the player enters an arena, joins a tournament or duels with a lord.

mtf_sit_day sets the music situation when it is daytime at the world map (not used in Native).

mtf_sit_encounter_hostile is not really getting used in Native, the track encounter_hostile_nords.ogg never plays.

mtf_sit_feast sets the music situation when a feast is going on in a lords hall which gets visited.

mtf_sit_fight sets the music sitation when a battle goes on.

mtf_sit_killed sets the music situation if the player got defeated in battle or is retreating.

mtf_sit_main_title sets the music at the main menu.

mtf_sit_multiplayer_fight sets the music at the multiplayer.[499]

mtf_sit_night sets the music situation when it is nighttime at the world map.

mtf_sit_siege sets the music situation when a siege is going on.

mtf_sit_tavern sets the music situation when visiting a tavern.

mtf_sit_town sets the music situation when visiting a town.

mtf_sit_town_infiltrate sets the music situation when infiltrating a town.

mtf_sit_travel sets the music situation when traveling on the world map.

mtf_sit_victorious sets the music situation if the player is victorius in battle.

mtf_start_immediately removes the delays between tracks and overrides song priority.[500]

Be aware that you can edit the usage of the already existing situation flags, they are not hard-coded (with the great exception of mtf_sit_main_title). There are 16 defined defined music situations in header_music.py but you can implement up to three more with the unused values between the mtf_sit_feast and mtf_module_track constants.[501] The flag mtf_sit_ambient_music is a prime example for this.

[499] MadVader, Which music tracks play in multiplayer, in Warband??.
[500] Swyter, Warband: Engine-Related Questions.
[501] Lav, Editing Music.

5.7.1.2 Culture Flags

There are six defined cultures (mtf_culture_1 to mtf_culture_6) with mtf_culture_all denoting all of them. As mentioned above, setting no culture flags is the same as putting mtf_culture_all. in original Vanilla M&B the sixth culture is used to denote a neutral faction, but is actually used for various bandits in the code. In Warband the sixth culture is used for Sarranids and, interestingly enough, still for various bandits. Note that in Native the player's party culture is 0, until he becomes a vassal. Examples for the usage of culture flags:

```
("ambushed_by_neutral", "ambushed_by_neutral.ogg", mtf_sit_ambushed|mtf_sit_siege,
    mtf_sit_fight)
```

This track Will play and continue in all cultures, mtf_culture_all is implied due to no culture flag being set.

```
("ambushed_by_neutral", "ambushed_by_neutral.ogg", mtf_culture_1|mtf_culture_2|
   mtf_sit_ambushed|mtf_sit_siege, mtf_culture_4|mtf_sit_fight)
```

Will start playing for cultures 1 and 2, will continue playing for cultures 1, 2 and 4 (i.e. will fade out in cultures 3, 5 and 6).

Take note that it is not possible to add new culture music flags like mtf_culture_7 as those six flags are hard-coded in the game engine. You can however control the music with operations (the ones affecting the music tracks are play_track, play_cue_track, music_set_situation, music_set_culture) and thus override the default music system to have it be based on some other in-game variable than the culture flags.[502]

Besides the vanilla setup, you may use the fact that Warband is picking music tracks based on the current culture/situation combination. Nothing prevents you from treating those cultures and situations as just abstract numbers. And instead of six cultures and 19 situations - remember, three more can be added to the 16 Native ones - you get 6*19 = 114 unique situations. Actually less as some situations are quite specific (for example mtf_sit_main_title) but you can still get a respectable number to create the needed variety.[503]

5.7.1.3 Native Music Code

Most of the code in the game changes the music through the script music_set_situation_with_culture. For scripters, it is a wrapper for music_set_situation and music_set_culture. For everybody else, it analyzes a *situation* (see mtf_sit_x flags) and finds one or a variety of cultures appropriate to the

[502] Somebody, Modding Q&A, and kalarhan, Modding Q&A. The DLC *Viking Conquest* and the Mod *The Last Days* have custom music systems for example.

[503] Lav, Editing Music.

situation. Multiple cultures will cause more matches found among the tracks, which will either add to the variety of music or confuse the player with seeming randomness. It is a thin line between both. This script only implictly plays music, the M&B internal music system detects changes in situation and culture and does the actual playing.

Examples of situations and cultures:

- In town (`mtf_sit_town`), set the culture to the original faction of the town.
- While travelling the map (`mtf_sit_travel`), set the culture to the faction of the nearest town AND the player's culture (two cultures max).
- In battle (`mtf_sit_fight` or `mtf_sit_ambushed`), set the culture to both the leading parties AND the faction of the nearest center (three cultures max).
- If victorius in battle (`mtf_sit_victorious`), the culture is player's own, if defeated in battle (`mtf_sit_killed`), the culture is the enemy's.
- In multiplayer, set the culture to both team factions (two cultures).[504]
- Some situations are not culture-specific (culture=0) - make sure you don't put `mtf_culture_x` play conditions with these in module_music.py: `mtf_sit_tavern`, `mtf_sit_siege`, `mtf_sit_arena`.
- Some situations are never used by the code, so don't waste your tracks with these: `mtf_sit_town` (Warband uses this), `mtf_sit_day`, `mtf_sit_night` (MnB: `calm_night_2`.ogg never plays, Warband uses this), `mtf_sit_encounter_hostile` (`encounter_hostile_nords`.ogg never plays).

Some common music situations (`mtf_sit_x`) as set in the code are Town menu (travel), Town streets (travel, Warband: town or night), Streets night (travel), Tavern (tavern), Arena Master (travel), Arena melee (arena), Village walk (travel), Castle menu (travel) and Castle hall (travel).

Battles are special as they use triggers to play music. The common battle trigger fires every 30 seconds and sets the situation to `mtf_sit_fight` or `mtf_sit_ambushed` if the odds against the player are worse than 2:1. Siege battles start with common siege music, then play battle music as normal.

The world map has a trigger that fires every game hour and tries to play travel music which sets the situation to `mtf_sit_travel`.

There are also some exceptions since some tracks are explicitely played from scripts, so they don't need any flags. Examples are for example the Native music tracks `captured`, `empty_village`, `escape`, `victorious_evil` and (only in Warband) `wedding`.

[504] Check `script_multiplayer_init_mission_variables` for more details.

5.7.1.4 Tips for Music Designers

You may want to have special tracks for every possible situation in the game - resist the urge! From a player point of view more varied travel and battle tracks may bring better value.

When deciding which tracks can continue where (continue condition flags) think in terms of what the average player is likely to do - if he spends only a few seconds in your taverns, maybe you could allow the travel or town or lord's hall music to continue there, so you will not change your music every few seconds; if your taverns are more interesting to hang out, do not allow anything to play there except tavern music.

Try to use `mtf_persist_until_finished` on shorter tracks only (like Native's victory tracks). If you want to annoy the player, maybe because he is a cheater, a pirate or you are just plain evil, set these flags:

```
("punish_player", "punish_player.ogg", mtf_looping|mtf_persist_until_finished|
    mtf_module_track, 0)
```

The track will loop forever and cannot be stopped until the player quits the game.

5.7.1.5 Tips for Scripters

If you want to let any current music fade out, use `(music_set_situation, 0)`. The new music will start playing when the situation is set again. Use the `play_track` operation when you want to play a specific music track and don't want to allow M&B to choose one randomly. Watch out for the continue flags or your music track may stop short. You can use the operation `play_cue_track` to play a music track over any already playing track. However, there couldn't be thought of a good use for this yet.

The simplest way to play a music track only for a specific encounter (e.g. a zombie horde) is done in following steps:

1. In `game_event_party_encounter()` add these lines:

```
(try_begin),
    (eq, "$g_encountered_party_template","pt_zombie_horde"),
    (play_track, "track_zombie_horde", 1),
(try_end),
```

2. Add a new track in module_music.py (and put `zombie_horde.ogg` in (mod dir)/Music):

```
(("zombie_horde", "zombie_horde.ogg", mtf_module_track, mtf_sit_fight|
    mtf_sit_ambushed|mtf_sit_travel|mtf_culture_all),
```

As mentioned above in Chapter 5.7.1.1 about the music track flags, you can add up to three more situation flags to the Native ones. The flag `mtf_sit_ambient_music` was already mentioned as a prime example for how one got added by the developers of VC. There was also a Native situation flag which got removed out of unknown reasons but for which you can still find remnants in the code. So for reimplementing lord's hall (castle and town castle hall) music you will need to follow these easy steps:

1. Add this line to header_music.py:[505]

```
mtf_sit_lords_hall                          = 0x02000000
```

2. Find both lines in module_mission_templates.py and uncomment them:

```
#           (call_script, "script_music_set_situation_with_culture",
    mtf_sit_lords_hall),
```

3. Add one line to `music_set_situation_with_culture` in module_scripts.py:

[505] Obviously you need to use `0x04000000` instead if you have also added `mtf_sit_ambient_music`.

```
(this_or_next|eq, ":situation", mtf_sit_town_infiltrate),
(this_or_next|eq, ":situation", mtf_sit_lords_hall),        <<<——
(eq, ":situation", mtf_sit_encounter_hostile),
```

4. Add tracks to module_music.py (change the flags as you like), and add .oggs to your mod's Music folder:

```
("lords_hall_swadian", "lords_hall_swadian.ogg", mtf_module_track|
    mtf_culture_1|mtf_sit_lords_hall, mtf_sit_tavern|mtf_culture_all),
("lords_hall_vaegir", "lords_hall_vaegir.ogg", mtf_module_track|
    mtf_culture_2|mtf_sit_lords_hall, mtf_sit_tavern|mtf_culture_all),
("lords_hall_khergit", "lords_hall_khergit.ogg", mtf_module_track|
    mtf_culture_3|mtf_sit_lords_hall, mtf_sit_tavern|mtf_culture_all),
("lords_hall_nord", "lords_hall_nord.ogg", mtf_module_track|mtf_culture_4|
    mtf_sit_lords_hall, mtf_sit_tavern|mtf_culture_all),
("lords_hall_rhodok", "lords_hall_rhodok.ogg", mtf_module_track|
    mtf_culture_5|mtf_sit_lords_hall, mtf_sit_tavern|mtf_culture_all),
("lords_hall_sarranid", "lords_hall_sarranid.ogg", mtf_module_track|
    mtf_culture_6|mtf_sit_lords_hall, mtf_sit_tavern|mtf_culture_all),
```

5.7.1.6 Optimal Music File Format

At music the file format seems to be not as important as at the sounds. You can use OGG, MP3 and WAV files as music tracks. The developers used the format OGG Vorbis because is free to use and doesn't include any patents[506] OGG Vorbis is a patent-free container with even better quality than the costly MP3 and the game fully understands it.[507] This file format has a backdraw at sounds which is why you should read also the subchapter about the optimal sound file format! However, testing indicates that music files can remain OGG; they are supposed to be in a streaming format and changing them to WAV is probably not a good idea.[508] Make sure your tracks are stereo, mono tracks might have some problems playing properly.

[506] Swyter, Added music doesn't play on time.

[507] Swyter, Modding Q&A.

[508] xenoargh, Optimization Announcement: Sound and your Mods.

5.7.2 Module Sounds

```
Each sound record contains the following fields:
1) Sound id: used for referencing sounds.
2) Sound flags: see header_sounds.py for a list of available flags.
3) Sound file: filename of the sound.
```

module_sounds.py is where the sound effects of the module are getting recorded and defined. As module_music.py before the file simply consists of a list of tuples. Sounds are getting used at a huge variety of occassions, like footsteps of troop and horse agents, yelling sounds of human agents and snorts of horses or clashes and releases of weapons, to name just a few of them.

5.7.2.1 Sound Flags

The following sound flags are available for usage:

sf_2d lets the sound's volume level not be depend anymore on its distance to the camera but be constant. 2d sounds should also only take stereo into account.

sf_always_send_via_network forces a sound run on server with the operation `agent_play_sound` to always be synced to all clients, even if they are more then 20 meters away from this agent (the default limit).[509]

sf_looping lets the sound loop.

sf_priority_x determines on a scale from 1 to 15 the priority of this sound. Higher priority sounds tend to overwrite lower ones if all sound channels are currently being used (at the same time). Distance also manipulates the resulting priority:[510]

```
# sound priority of 0-256 (0 hightest 256 lowest) calculated as such:
# float pf = (base_dist / (base_dist + dist)) * ((1.0f - priority_effect) + priority_effect * (priority_squared ? get_priority() * get_priority() : get_priority()));
# sounds prio lower then 225 override other sounds in the list with lowest priority.
sound_base_dist = 3.5
sound_priority_effect = 0.45
sound_priority_squared = 1
log_sound_priority = 0
```

[509] Vincenzo, Version 1.150 - 1.151 - 1.152 - 1.153 released.

[510] _Sebastian_, Modding Q&A.

sf_start_at_random_pos lets the sound start at a random position in the sound file.

sf_stream_from_hd lets this sound be streamed from harddisk. Recommended for large sound files that are not played often.

sf_use_next_for_far WFaS sound flag. Up4tests if working in Native as well.

sf_vol_x determines on a scale from 1 to 15 the volume of a sound, the higher the value the higher the level of volume.

5.7.2.2 Optimal Sound File Format

Most of the knowledge about this topic is based on many tests of xenoargh who was afterwards fairly confident that he has pinpointed the problem of seemingly random lockups that would halt the game state, and found a solution. It turned out that problem isn't graphics at all; HERR_BUFFER_LOCK errors were a symptom of CPU lock states happening, not the result of running out of VRAM. The real lockup issue appears to be sounds. In particular, the sound format. He converted all of the OGG files (including all of the Taleworlds sounds) to 16-bit 22kHz mono WAV and the HERR_BUFFER_LOCK errors disappeared.[511]

As for music tracks you can in theory use OGG and WAV (and MP3) files for the sound effects.[512] Warband is however known to have problems with sounds. The sound engine used is performing much better with WAV files. If you have only OGG files, these will perform worse and crash the engine sooner or later. All short files should be WAV! Using WAV files might increase the overall filesize of the DLC but in the end you are going to profit from it.[513]

It is a common misconception that OGG files are smaller but they are only smaller on disk. As soon as the game engine starts up it will unpack all these OGG files and put them into memory. Your 6kb large OGG files will become 10mb.[514] Warband just reuses the reference decompressor libraries which are free. However, they use them probably inefficiently, because the game engine is the one having to implement the streaming part with the generic tools and FMOD (which is the middleware they use). Storing sounds as WAV might seem pretty wasteful, albeit sounds and music are still a

[511] xenoargh, Optimization Announcement: Sound and your Mods. It should however be noted that late posts at this thread as well as at the thread Testing a possible optimization: all sounds converted to .wav format that using .wav files instead of .ogg doesn't increase performance with Warband 1.143 (as opposed to 1.134 at which xenoargh did his tests). cmpxchg8b notes however that a great majority of players use on-board sound cards, which offload as much processing as possible to the CPU and thus cause the CPU bottleneck. Therefore the here reported findings hold still up at the most recent stable game version 1.174.

[512] Swyter, Added music doesn't play on time.

[513] Death by EMP posting for an anonym person, The REAL reason why this DLC is performing bad and crashing - Including Tips.

[514] Death by EMP posting for an anonym person, The REAL reason why this DLC is performing bad and crashing - Including Tips.

small part of the mod package. However, if the CPU is bottlenecked in Warband due to all the stuff going on and the game engine doesn't use extra threads/CPU cores for decoding/uncompressing Vorbis/MP3 sound, then it makes sense that doing that extra processing every frame adds some overhead which the WAV format doesn't have. This is just educated guessing.[515]

Contrary, large sound files that are not played often should be streamed from harddisk. This will mean they are not loaded into memory at the startup and will not strain the memory so much. These sound files should be in OGG to not get in trouble with harddisk slowness. Use the sound flag `sf_stream_from_hd` at the respective entries in your module_sounds.py.[516]

5.7.2.3 Other notes

Open question about upper limit at sounds.[517], answered here[518]

Different death sounds[519]

sf_priority[520]

Sounds info bits[521]

extra faction tuple in sound entries at WFAS[522]

sounds for different kinds of horses[523]

2D determines just plays stereo with left and right determining volume in each channel, without falloff from distance. 3D determines the volume by distance from listener, with the left and right channels being determined by listener facing. Native obviously supports 3D audio. (dstn)

Silencing horses[524]

limit of sounds to a weapon[525]

32 samples per sound[526]

sounds at distance[527]

sound priority[528]

bullet sounds[529]

[515] Swyter, Mount & Blade Modding Discord.

[516] Death by EMP posting for an anonym person, The REAL reason why this DLC is performing bad and crashing - Including Tips.

[517] Caba‘drin, Modding Q&A.

[518] MadVader, Modding Q&A.

[519] Somebody, Modding Q&A.

[520] MadVader, Modding Q&A.

[521] Caba‘drin, Modding Q&A.

[522] Somebody, Modding Q&A.

[523] MadVader, Modding Q&A.

[524] Lumos, Modding Q&A.

[525] kalarhan, Modding Q&A.

[526] _Sebastian_, Modding Q&A.

[527] _Sebastian_, Modding Q&A.

[528] _Sebastian_, Modding Q&A.

[529] kalarhan, Modding Q&A and Modding Q&A.

speed of sound simulation script[530]

Sound modification[531]

[530] _Sebastian_, Modding Q&A.

[531] Colt, Sound Modification Guide.

5.8 VFX-related Module System Files

5.8.1 Module Particle Systems

```
Each particle system contains the following fields:
 1) Particle system id (string): used for referencing particle systems in other files.
    The prefix psys_ is automatically added before each particle system id.
 2) Particle system flags (int). See header_particle_systems.py for a list of available flags
 3) Name of the mesh.
 4) Num particles per second: Number of particles emitted per second.
 5) Particle Life: Each particle lives this long (in seconds).
 6) Damping: How much particle's speed is lost due to friction.
 7) Gravity strength: Effect of gravity. (Negative values make the particles float upwards.)
 8) Turbulance size: Size of random turbulance (in meters)
 9) Turbulance strength: How much a particle is affected by turbulance.
10,11) Alpha keys : Each attribute is controlled by two keys and  each key
12,13) Red keys   : has two fields: (time, magnitude). For example
14,15) Green keys : scale key (0.3,0.6) means scale of each particle
16,17) Blue keys  : will be 0.6 at the time 0.3 (where time=0 means
18,19) Scale keys : creation and time=1 means end of the particle)
The magnitudes are interpolated in between the two keys and remain constant beyond the keys.
Except the alpha always starts from 0 at time 0.
20) Emit Box Size : The dimension of the box particles are emitted from.
21) Emit velocity : Particles are initially shot with this velocity.
22) Emit dir randomness
23) Particle rotation speed: Particles start to rotate with this (angular) speed (degrees per second).
24) Particle rotation damping: How quickly particles stop their rotation
```

Some intro text here

5.8.1.1 Particle System Flags

psf_always_emit marks particle systems which are always emitting.

psf_global_emit_dir

psf_emit_at_water_level

psf_billboard_2d

psf_billboard_3d

psf_billboard_drop

psf_turn_to_velocity

psf_randomize_rotation

psf_randomize_size

psf_2d_turbulance

psf_next_effect_is_lod

5.8.1.2 Other notes

Links to posts which contain informations about particle systems, need to integrate them at a later stage: fisheye. Modding Q&A; bryce777, Modding Q&A; fisheye, Modding Q&A; Somebody, Modding Q&A; ithilienranger, Modding Q&A, Modding Q&A; cmpxchg8b, Modding Q&A; Somebody, Modding Q&A; cmpxchg8b, Modding Q&A; Patta, Modding Q&A; Vornne, Modding Q&A; Some particle system discussion here and here; Vornne and cmpxchg8b, here;

Particle system values[532]

particle limitations[533]

particle stuff[534]

particle effects [535]

particle system only shows up at night[536]

[532] jacobhinds, Modding Q&A.

[533] jacobhinds, Modding Q&A, and Somebody, Modding Q&A.

[534] Docm30, Modding Q&A.

[535] _Sebastian_, Modding Q&A.

[536] Abhuva (credit), Nordous' Sceners Guild.

5.8.2 Module Postfx

skybox postfx stuff[537]

[537] Vornne, Modding Q&A and Modding Q&A.

5.8.3 Module Tableau Materials

Tableau material can use arbitrary tableau meshes, with different textures.[538]

Heraldic items submeshes affected too[539]

banner background colour array[540]

clickable tableaus[541]

character screens[542]

heraldic boots[543]

gloves and boots no heraldic tableau[544]

No submeshes for heraldic items [545]

parameter tableau material[546]

full helmet taken off at tableau[547]

[538] GetAssista, Modding Q&A.

[539] oliver, Mount & Blade Modding Discord.

[540] MadVader, Modding Q&A, and _Sebastian_, Heraldic Armor issue [help].

[541] Somebody, Modding Q&A.

[542] The_dragon, Modding Q&A.

[543] Discussion, Modding Q&A.

[544] Somebody, Modding Q&A.

[545] NPC99, Modding Q&A.

[546] cmpxchg8b, [WB] Warband Script Enhancer v3.2.0.

[547] Somebody, Modding Q&A.

6 Documentation module.ini

INI, short for initialization, is a Microsoft standard for the initialization and configuration of programs. Use a basic text-editor such as Notepad for the INI file. This file is always local and modularized and contains various mod configuration settings and tells the game which BRFs to load. Following options in module.ini can be set (sorted in categories; marked with VC if introduced with Viking Conquest, NW with Napoleonic Wars):

6.1 Module General Parameters

6.1.1 Versions

module_name = Calradia string, is used for naming your module.

module_version = 0 int, can be used for multiplayer and single player (saved game) module versioning.

compatible_module_version = 0 int, can be used for multiplayer module versioning.

compatible_multiplayer_version_no = 1170 int, current multiplayer version, used with **compatible_module_version** to determine if a player can connect to a server.

compatible_savegame_module_version = 0 int, can be used for single player (saved game) module versioning.

compatible_with_warband = 1 boolean, is most probably getting used to mark original M&B mods that they can get played with M&B Warband too.

works_with_version_max = 1011 deprecated, int, use compatible_savegame_module_version and module_version instead.

works_with_version_min = 1000 deprecated, int, use compatible_savegame_module_version and module_version instead.

6.1.2 Loading Resources

Outside of module configuration options, **module.ini** is also used to tell the engine what **.BRF** resources to load, and where to find them.

load_resource = resource_name Tells the engine that this resource **(.BRF)** exists in either the ***/CommonRes/** or ***/(module-name)/Resources/** folders. It doesn't seem to add to a much longer load times, but it is more inefficent than telling the engine where to look.

load_resource_nofast = resource_name same as load_resource, no difference.

load_mod_resource Explicitly tells the engine that this resource **(.BRF)** exists inside ***/(module-name)/Resources/** folder. It should result in faster load times.

load_module_resource Alternate syntax for **load_mod_resource**.

scan_module_sounds = 0 boolean, determines if the engine will scan the module's ***/Sounds/** folder.

scan_module_textures = 0 boolean, determines if the engine will scan the module's ***/Textures/** folder.

use_case_insensitive_mesh_searches = 0 boolean, determines if your module allows case insensitive references to mesh names i.e. **sword_a** and **Sword_a** being treated as the same reference.

6.1.3 Enabling Content

has_custom_battle = 0 boolean, determines if Custom Battle appears on the main menu.

has_multiplayer = 0 boolean, determines if Multiplayer appears on the main menu.

has_single_player = 1 boolean, determines if New Game and Load Game appear on main menu.

has_tutorial = 0 boolean, determines if Tutorial appears on the main menu.

enable_quick_battles = 1 boolean, determines if Quick Battles appears on the main menu. Quick Battles is the old 1.011 version of Custom Battles, so this flag works only at Vanilla M&B.

6.2 World Map Parameters

6.2.1 Map Constants

map_max_distance = 175.0 float, maximum amount of zoom out on the world map, maximum seems to be 251.

map_min_x = -180 int, determines the eastern bounds of the world map. The map can continue past this, but parties will not be able to travel past this boundary.

map_max_x = 180 int, determines the western bounds of the world map. The map can continue past this, but parties will not be able to travel past this boundary.

map_min_y = -145 int, determines the northern bounds of the world map. The map can continue past this, but parties will not be able to travel past this boundary.

map_max_y = 145 int, determines the southern bounds of the world map. The map can continue past this, but parties will not be able to travel past this boundary.

map_min_elevation = 0.2 float, sets the minimum distance the camera can be from the terrain in the world map. Controlls the angle too.

map_max_elevation = 1.0 float, sets the maximum distance the camera can be from the terrain in the world map.

map_river_direction = 140 int, angle used to determine the flow direction of the river shader for the world map.

map_river_speed_x = 0.01 float, used to determine the flow speed on the x-axis of the river shader for the world map.

map_river_speed_y = -0.01 float, used to determine the flow speed on the y-axis of the river shader for the world map.

map_sea_direction = -40 int, angle, sea wave foam direction.

map_sea_wave_rotation = 300 angle, module.ini comment says "This is where the tear artefact is visible on the sea."

map_sea_speed_x = 0.02 float, used to determine the flow speed on the x-axis of the ocean shader for the world map.

map_sea_speed_y = -0.02 float, used to determine the flow speed on the y-axis of the ocean shader for the world map.

Just like at the world map terrain the map_trees material of map tree models is hard-coded, no matter what you set in the respective brf file.

map_tree_types = 17 int, is the number of tree types that appear on the normal terrain, using the meshes map_tree_a through map_tree_r on for map trees. Equals the amount of forest tree models the map will randomly cycle through.

map_desert_tree_types = 4 int, is the number of tree types that appear on the desert terrain, equals the amount of desert tree models the map will randomly cycle through.

map_snow_tree_types = 3 int, is the number of tree types that appear on the snow terrain, equals the amount of snow tree models the map will randomly cycle through.

map_steppe_tree_types = 5 int, should be the number of tree types that appear on the steppe terrain, equals the amount of steppe tree models the map will randomly cycle through.

6.2.2 Time and Date Parameters

starting_day Deprecated, int, use game_get_date_text instead.

starting_month Deprecated, int, use game_get_date_text.

starting_year Deprecated, int, use game_get_date_text.

time_multiplier = 0.25 float, speed with which time passes on the map screen. With 0.25 as default value 1 GAME hour is 4 seconds of REAL time, so in-game time passes 4 times slower than in real life (each hour in-game is equal to 1 second real life divided by the value of the time_multiplier). A value of 2.4 thus means that 1 in-game day takes 10 seconds IRL (or 1 second via CTRL+SPACE).

6.2.3 Leveling-related Parameters

player_xp_multiplier = 2.0 float, sets the xp multiplier the player gains by killing an enemy (max 10, perhaps 15).

hero_xp_multiplier = 2.0 float, sets the xp multiplier heroes gain by killing an enemy (max 10, perhaps 15).

regulars_xp_multiplier = 3.0 float, sets the xp multiplier regular troops gain by killing an enemy (max 10, perhaps 15).

level_boundary_multiplier = 2.0 VC, float, sets the multiplier for EXP needed for troop to upgrade.

attribute_points_per_level = 0.195 VC, float, sets how many attribute points are gained per level, 0.125 in Native(?).

attribute_required_per_skill_level = 2 VC, int, sets how many attribute points are required per skill level, 3 in Native. According to Vjeffae the in-game maximum skill level is equal to the current base attribute level divided by this value (rounded down).

skill_points_per_level = 2 VC, int, sets how many skill points are gained per level.

weapon_points_per_level = 5 VC, int, sets how many weapon proficiency points are gained per level, 10 in Native.

can_run_faster_with_skills = 0 boolean, determines how agility and athletics affect running speed. Upfollowing engine formula is given for it:

```
if (rglConfig::Battle::bCanRunFasterWithSkills)
troopFactor = ((agility + athletics * 6.0f + 25.0f) * 30.0f / (weightFactor + 30.0f) + 90.0f) / 100.0f;
else
troopFactor = ((agility * 0.7f + athletics * 3.0f + 25.0f) * 70.0f / (weightFactor + 70.0f) + 90.0f) / 100.0f;
```

Written down for better readability:

```
if can_run_faster_with_skills = 1
troopFactor = ((agility + athletics * 6 + 25) * 30 / (weightFactor + 30) + 90) / 100;
else
troopFactor = ((agility * 0.7 + athletics * 3 + 25) * 70 / (weightFactor + 70) + 90) / 100;
```

player_wounded_treshold = 5 int, determines the minimum number of hitpoints a player must have to appear in battles and contribute to party skills.

hero_wounded_treshold = 15 int, determines the minimum hitpoints needed for heroes to appear in battles and contribute to party skills.

base_companion_limit = 20 int, maximum amount of companions in the player party without leadership skill.

skill_leadership_bonus = 3 int, sets how many additional troops per leadership skill point the player can have in his party.

skill_prisoner_management_bonus = 5 int, sets how many additional prisoners per prisoner management skill point the player can have in his party.

track_spotting_multiplier = 0.8 float, multiplier of the tracking skill. If set to 0 no tracking marks will appear at the world map no matter the skill.

6.2.4 Item Parameters

display_wp_archery = 0 boolean, determines whether or not items with the **itp_type_bow** flag will appear in the module.

display_wp_crossbows = 0 boolean, determines whether or not items with the **itp_type_-crossbow** flag will appear in the module.

display_wp_firearms = 0 boolean, determines the visibility of the weapon proficiency **Firearms** inside the character menu, but the weapons and the ability to use them still exists in the game. You can actually level them up, even if it is marked 0.

display_wp_one_handed = 0 boolean, determines the visibility of the weapon proficiency **One Handed** inside the character menu, but the weapons and the ability to use them still exists in the game. You can actually level them up, even if it is marked 0.[548]

display_wp_polearms = 0 boolean, determines the visibility of the weapon proficiency **Polearms** inside the character menu, but the weapons and the ability to use them still exists in the game. You can actually level them up, even if it is marked 0.

display_wp_throwing = 0 boolean, determines the visibility of the weapon proficiency **Throwing** inside the character menu, but the weapons and the ability to use them still exists in the game. You can actually level them up, even if it is marked 0.

display_wp_two_handed = 0 boolean, determines the visibility of the weapon proficiency **Two Handed** inside the character menu, but the weapons and the ability to use them still exists in the game. You can actually level them up, even if it iss marked 0.

timid_modifier_speed_bonus = 0 float, the speed modifier for the horses that have the **timid**-modifier.

meek_modifier_speed_bonus = 0 float, the speed modifier for the horses that have the **meek**-modifier

[548] All display_wp_* settings tested by dstn.

use_crossbow_as_firearm = 0 boolean, mostly depreciated, used for a workaround in WFaS or older mods where **itp_crossbow** was used instead of **itp_musket** and **itp_pistol**.[549] Determines whether crossbows will fire with smoke and a sound of explosion or with normal crossbow sound.[550]

6.2.5 Party Parameters

auto_compute_party_radius = 1 VC, boolean, uses the party icon model to determine party radius. Especially important for modules with large town icons as it allows parties to interact with them in expected locations, as opposed to the center of the icon. Without it icons will disappear when their centre is off screen even though they should still be visible. For extremely large icons it fails to avoid 100 % of visual problems as the camera changes perspective. You need to test large icons in-game to ensure they are not too large.

seeing_range = 6.5 float, determines the default seeing range of a party on the world map, the visibility radius.

show_party_ids_instead_of_names = 0 boolean, determines if a party's **id** value is show above them instead of their name. Change to 1 for ease in module development.

use_strict_pathfinding_for_ships = 1 VC, boolean, keeps ships at sea for sea-to-sea routes.

6.2.6 Game Menus Parameters

show_troop_upgrades_button = 0 VC, boolean, determines if the mod shows the troop upgrade button.

6.2.7 Others

show_faction_color = 1 boolean, determines if party names on the world map will show their faction color (1) or not (0). In latter case the names appear in a sort of brownish-white colour.

show_quest_notes = 1 boolean, determine if notes appear on the quest screen.

has_accessories_for_female = 0 boolean, allows the mod to use the beard slot in character creation for accessories on female troops such as jewelry.[551] No examples are found in Native.

[549] Alternatively one can also add a new firearm category by replacing all crossbow related sounds, animations and proficienies as needed. Some notes are given here by HokieBT.

[550] Baskakov_Dima, Modding Q&A.

[551] _Sebastian_, has_accessories_for_female?.

disable_disband_on_terrain_type = 0 VC, boolean, used to insure parties cannot disband on certain tiles (Like passable water).

disable_force_leaving_conversations = 1 VC, boolean, disables the TAB end in conversations.

can_adjust_camera_distance = 1 boolean, determines if the player can use numpad + and numpad - to adjust the default camera position.

limit_hair_colors = 1 boolean, set to 1 if you don't have extra hair textures. Each human skin definition in Native uses hair_blonde along with vertex coloring. If you set the value to 0 the game expects to find for example a material and texture named hair_red, beard_black, etc.

auto_create_note_indices = 0 boolean, determines whether or not automatically search through all troops/factions/towns to check if they have note text.

6.3 Battle Scene Parameters

6.3.1 Combat Parameters

According to module.ini comments all missiles with damage > shield_penetration_offset + shield_penetration_factor * shield will penetrate.

shield_penetration_offset = 30.0 float, missiles with (damage > shield_penetration_offset + shield_penetration_factor * shield) will penetrate a shield and cause damage to the wielder.

shield_penetration_factor = 3.0 float, used in conjunction with shield_penetration_offset to determine if a missile can penetrate a shield.

crush_through_treshold = 2.4 float, determines how much damage is required to crush through overhead blocks on weapons with the appropriate flag **itp_crush_through**.

According to module.ini comments you can modify the damage system by editing the following values: The first three values (i.e. soak) determine the amount which will be directly subtracted from damage due to armor. The next three values (i.e. reduction) determine the percentage reduction from the damage. The same goes for extra_penetration factor. This counts only for weapons that have the flag **itp_extra_penetration**. Also, the *soak* can be negative, meaning that it will actually ADD to the damage, Not sure about the *reduction*, though in theory it should work the same way, but less violently.

armor_soak_factor_against_cut = 0.8 float, determines the amount which will be directly subtracted from cut damage due to armor.

armor_soak_factor_against_pierce = 0.65 float, determines the amount which will be directly subtracted from pierce damage due to armor.

armor_soak_factor_against_blunt = 0.5 float, determines the amount which will be directly subtracted from blunt damage due to armor.

armor_reduction_factor_against_cut = 1.0 float, determines the percentage reduction from the cut damage.

armor_reduction_factor_against_pierce = 0.5 float, determines the percentage reduction from the pierce damage.

armor_reduction_factor_against_blunt = 0.75 float, determines the percentage reduction from the blunt damage.

extra_penetration_factor_reduction = 1.0 float, used in conjunction with **itp_extra_penetration** to determine at what level extra penetration is reduced by armor. In Native no extra penetration flags are set, so set to '1.0' to keep them ineffective.

extra_penetration_factor_soak = 1.0 float, used in conjunction with **itp_extra_penetration** to determine at what level extra penetration is soaked by armor. In Native no extra penetration flags are set, so set to '1.0' to keep them ineffective.

According to module.ini comments damage below this threshold levels will not interrupt melee attacks.

damage_interrupt_attack_threshold = 3.0 float, damage below this will not interrupt melee attacks.

damage_interrupt_attack_threshold_mp = 1.0 float, damage below this will not interrupt melee attacks in multiplayer.

ai_decide_direction_according_to_damage = 1 boolean, when set to 1, the AI will take into consideration what each directional attack's damage will be when deciding what attack to use.

apply_all_ammo_damage_modifiers = 1 VC, boolean, allows all ammo to apply Blunt/Sharp/Cutting modifier damage.

brace_rotation_limit = 0.012 float, affects items with the flag itp_is_pike and is used for rotation speed limiting while bracing spears (e.g. in NW). If the absolute turn amount is less than 0.01 or the set value (whichever is higher) the game engine plays the pike crouch animation (anim_crouch_pike) and otherwise the crouch staff animation (anim_crouch_staff). You can obviously not give a value below the default one of 0.01 and if you provide a big value (it seems like 0.5 is enough) you can completely override the canceling effect.

couched_lance_damage_multiplier = 0.65 float, determines how much additional damage a couched lance will do. (Native 0.65).

disable_attack_while_jumping = 0 boolean, determines if an agent can attack while jumping and if attacks should be interrupted when the agent starts jumping.

disable_zoom = 0 boolean, determines whether or not the player can use 'shift' to narrow the camera's field of vision.

fall_damage_multiplier = 1.0 float, used to determine how much damage is applied from falling. Airtime is otherwise probably the deciding factor for the falling damage.[552]

horse_charge_damage_multiplier = 1.0 float, used to scale the charge damage inflicted by horses when ramming into human agents.

lance_pike_effect_speed = 3.0 float, determines the necessary speed of the horse at which the pike causes the anim_horse_rear animation.

melee_damage_speed_power = 2.0 float, setting speed_power to 2.0 makes damage scale with the square of weapon's speed. You can set it to 1.0 to make it scale linearly.

missile_damage_speed_power = 1.9 float, setting speed_power to 2.0 makes damage scale with the square of missile's speed. You can set it to 1.0 to make it scale linearly.

no_friendly_fire_for_bots = 0 boolean, determines if AI agents in Mutiplayer can cause friendly fire.

6.3.2 Battle Parameters

battle_size_min = 150 VC, int, minimum battle size for module.

battle_size_max = 750 VC, int, maximum battle size for module.

[552] Somebody, Modding Q&A.

far_plane_distance = 5000 float, the amount of units (in centimeters) that the render plane extends out from the camera inside of scenes, default is 1250. It's the maximum distance (in meters) in which an object will render for an user. Enlarge this value if you want to use bigger/better/longer border terrains and have the player be able to view them.

6.3.3 Physics Parameters

air_friction_arrow = 0.002 float, allows us to set the drag coefficient. 0.002 is Native.

air_friction_bullet = 0.002 float, allows us to set the drag coefficient, but specifically on missiles shot by firearms. 0.002 is Native.

6.3.4 Others

consider_weapon_length_for_weapon_quality = 1 VC, boolean, makes agents take into account a weapon's length when choosing a weapon at their inventar.

disable_moveable_flag_optimization = 0 boolean, assign '1' for moveable physics on all scene props; no **sokf_moveable** flag will be needed.[553]

mission_object_prune_time = 180 int, seconds required before an scene_prop like a fired arrow or a dropped item is removed from the scene. Setting it to 0 will prevent the items from despawning during a mission which is however not recommend for performance reasons. Note that this wont apply to items which are spawned via code.[554]

6.4 Other Parameters and Details

6.4.1 Enabling Features

can_crouch = 0 boolean, determines if human agents can crouch. You will need to add the string `ui_crouch|Crouch:` at your ui.csv to make it displayed correctly at the Gamekey config menu.[555]

can_objects_make_sound = 0 boolean, determines if scene_props can emit sounds inside a scene.

[553] Setting that flag to 1 prevented an usual 1.143 game version crash. For more details read up at Siege Camp Icon.

[554] _Sebastian_, Modding Q&A.

[555] dunde, Modding Q&A.

can_reload_while_moving = 0 boolean, determines if agent can reload crossbows while running or walking. Does not affect horseback. Needs `use_crossbows_as_firearms` to be enabled as well.[556]

can_use_scene_props_in_single_player = 0 boolean, determines if singleplayer agents can interact with scene_props like in multiplayer (using doors and drawbridges).

disable_food_slot = 1 boolean, mostly depreciated, can be changed to 0 if you want to keep the food slot in inventory window. Used in very old versions of M&B (still working in Warband though!) to choose what food item the party can eat. Putting a horse into it lets the horse be slaughtered.

has_forced_particles = 0 boolean, allows the mod to 'force' particles. Probably used in instances where turning off particle effects would give undue benefit to a player, such as thick musket smoke.

horses_rear_with_attack = 1 boolean, determines if horses rear upon being struck with enough damage.

horses_try_running_away = 0 boolean, determines if riderless horse agents will try to get away from aggressive agents.

multiplayer_walk_enabled = 0 boolean, determines if the players on a multiplayer server can use 'shift' to walk instead of run. Important for NW style line battles. **disable_zoom** needs to be set to 1, otherwise zoom takes precedence at 'shift' (a zooming player always walks, you just see it with zoom disabled).

num_hints = 12 int, total number of hints visible on the secondary loading screen.

shorter_pistol_aiming = 1 VC, boolean, adds pistols to the exception just like in crossbow and muskets.

use_advanced_formation = 0 boolean, determines if the module uses Native's advanced formations.[557] Those add extensions to the order menus related to formations and weapon usage. Players can then let the troops line up from one row to five rows and order their archers to fire at their command. Some lines need to be added to ui.csv for this.[558] For the menu order these are:[559]

[556] DarthMongol, Mount & Blade Modding Discord.

[557] A 2 is perhaps possible too, up4research; Baskakov_Dima (credit), Module.ini stuff and Landowning?.

[558] dunde, Modding Q&A.

[559] Taken from reddit.

- `ui_order_form_1_row|Form one row`
- `ui_order_form_2_row|Form two rows`
- `ui_order_form_3_row|Form three rows`
- `ui_order_form_4_row|Form four rows`
- `ui_order_form_5_row|Form five rows`
- `ui_order_button_fire_at_my_command|Fire at my command`
- `ui_order_button_all_fire_now|All, Fire now`
- `ui_order_button_left_fire_now|Left, Fire now`
- `ui_order_button_middle_fire_now|Middle, Fire now`
- `ui_order_button_right_fire_now|Right, Fire now`
- `ui_order_button_weapon_usage_orders|Equipment orders`
- `ui_order_button_use_melee_weapons|Use melee weapons`
- `ui_order_button_use_ranged_weapons|Use ranged weapons`

and for the shouting order display:

- `ui_form_1_row_e_|%s, Form one row`
- `ui_form_2_row_e_|%s, Form two rows`
- `ui_form_3_row_e_|%s, Form three rows`
- `ui_form_4_row_e_|%s, Form four rows`
- `ui_form_5_row_e_|%s, Form five rows`
- `ui_fire_at_my_command_e_|%s, Fire at my command`
- `ui_all_fire_now_e_|%s, fire now`
- `ui_left_fire_now_e_|%s, left side fire`
- `ui_middle_fire_now_e_|%s, middle fire`
- `ui_right_fire_now_e_|%s, right side fire`
- `ui_use_melee_weapons_e_|%s, use melee weapons`
- `ui_use_ranged_weapons_e_|%s, use ranged weapons`

use_phased_reload = 0 boolean, determines if reloading for muskets and pistols can be done in steps. That means if your reload has been interrupted, you'll start from the begining of the phase at which the reload was stopped. Phased reload works if you use itcf_reload_musket or itcf_reload_pistol for the ranged weapon.

6.4.2 Graphical Parameters

add_set_neighbors_to_tangent_flag_to_shader = 1 boolean, Shader Parameter. Appears to pass along information about neighboring faces tangents. No examples are found in Native.

blood_multiplier = 6.0 VC, float, determines the amount (and size?) of blood particle effects.

disable_high_hdr = 0 boolean, if set to '1' disables high HDR effects in the module.

fix_gamma_on_dx7_operation_colors = 1 boolean, Shader Parameter, used for set_startup_ambient_light and set_startup_ground_ambient_light operations. Below decompiled code:

```
if ( !rgl_HLSL_mode && dword_AA35C4C )
{
*(float *)v2430 = a2;
v1304 = sqrtf(*(float *)v2430);
v1305 = cur_game;
*(float *)(v2460 + 408) = v1304;
*((float *)v1305 + 103) = sqrtf(*((float *)v1305 + 103));
v1306 = cur_game;
a2 = sqrtf(*((float *)cur_game + 104));
v1298 = cur_mission;
LODWORD(v2460) = cur_game;
*((float *)v1306 + 104) = a2;
}
```

use_bordered_shadow_sampler = 1 boolean, Shader Parameter. Unused by engine.

screenshot_format = 1 boolean, allows modders to set the screenshot file format when pressing Ctrl + Insert which are getting saved in .../ Mount&Blade Warband/Screenshots/<Name of Module>. Set 0 for jpg (default), 1 for png and 2 for bmp.[560]

6.4.3 Multiplayer Parameters

restrict_attacks_more_in_multiplayer = 0 boolean, if an agent is stabbing or doing an overhead attack then the limit of the rotational speed will either be set to 11 (if set to 1) or 16 (if set to 0, default). This forces a more restricting attacking method in multiplayer and should be able to keep spamming and feinting to a minimum.

show_multiplayer_gold = 0 boolean, determines if the player's gold is shown in multiplayer. Only the gold amount won't get displayed, the gold icon itself will still be present.

[560] The screenshot function seems to be broken when playing with Linux; Noxbru, Mount & Blade Modding Discord.

sync_block_directions = 0 boolean, determines if block directions are synced between client and server.

sync_ragdoll_effects = 0 boolean, determines if ragdolls are synced between client and server.

6.4.4 Performance and Logs

dont_supress_initial_warnings = 1 boolean, should be set to 1 for debugging purposes and 0 for public releases. It should be used to hide loading warning for modules in the rgl_log.txt like the following one:

```
rgl_post_warning_line: WARNING: Unable to find material for mesh
    prt_mod_weapon_aim
```

give_performance_warnings = 1 boolean, gives warnings of potentially performance intensive scenes. Edit Mode should be enabled for this (Earendil). Warns with a message that performances may be degraded because of settings (Vjeffae). Up for research.

maximum_number_of_notification_messages = 4 VC, int, number of messages in battle, common are 10.

reduce_texture_loader_memory_usage = 0 boolean, should improve loading time by making it multithread.

supports_directx_7 = 1 boolean, determines if your module supports directx7 mode. Necessary for very old PCs.

use_scene_unloading = 1 VC, boolean, once you leave the scene, it's unloaded from your computer RAM. If you want to go back to this scene quickly, you will have to load it again. Loading of frequently visited scenes will be slower but it will save some of your RAM.

use_texture_degration_cache = 0 boolean, should improve loading times but generates cache files. When you tab out for too long, the game de-loads most of the textures and replaces them with those microtextures. This was the case in earlier versions too but the game didn't save them to the disk.[561]

6.4.5 Savegames

dont_load_regular_troop_inventories = 1 boolean, determines if the game saves inventories for regular troops (i.e. not heroes) too. Also, savegames will be a bit bigger and may thus crash.

[561] jacobhinds, Modding Q&A.

7 Game Engine Settings, Formulas and Info Tidbits

7.1 Remarks regarding txt Files

7.1.1 games_variables.txt

Cannot add variables in game_variables.txt[562]

There are also a couple of game options which didn't make it into 1.168(Steam version) by default. Compare also with WFaS and VC and create new file[563]

7.1.2 variables.txt

Global variable names are not getting used by the game engine, variables.txt is not even getting loaded by the engine and its only purpose is savegame compatibility. If you check the compiling system of the Module System you will notice that it preserves the order in variables.txt, to make sure that variable x stays in position y even if the declaration order is changed or variables have been added or removed, to keep savegame compatibility.[564]

It can happen that you compile and keep getting the messages "*variables.txt not found. Creating new variables.txt file*" and "*variable_uses.txt not found. Creating new variable_uses.txt file*" even though these txt files exist in your compile export folder. This is only a problem if you care about save game compatibility. Copy in that case working copies from your module folder to your source folder (same as the . py files). Otherwise the globals won't keep the same ID and mess with old saves.[565]

7.1.3 variable_uses.txt

variable_uses.txt is a file which doesn't matter a lot. It is there to keep savegame compatibility and is auto-generated by the Module System, used during the compilation phase and then no more. Warband doesn't know about it. Editing it by hand is wrong and does nothing at all.[566]

Integrate comment here[567]

[568]

[562] cmpxchg8b, Modding Q&A.
[563] _Sebastian_, Warband Modder's Download Repository.
[564] cmpxchg8b, Modding Q&A.
[565] kalarhan, Modding Q&A.
[566] cmpxchg8b, Modding Q&A, Modding Q&A and Modding Q&A.
[567] kalarhan, Modding Q&A.
[568] cmpxchg8b, Modding Q&A.

7.1.4 rlg_log.txt

semi-crash log[569], need to explain it a bit more detailed.

rgl_log is in your installation directory (one level above your Modules folder, for instance). (C:\Program Files\Mount&Blade Warband\ or something like it)[570]

7.1.5 rlg_config.txt

rgl_config is in your savegame/settings directory. (C:\Users\Username\Documents\Mount&Blade Warband\ or something like it)[571]

7.2 Damage System

7.2.1 Hit Detection

The game engine checks the collision between one of target's hitboxes with the attacker's right item bone. For latter it uses a capsule attached to the right item bone.[572]

7.2.2 Damage Formula[573]

```
if hold_time >= 1.1
        hold_bonus = 1.2
elif hold_time >= 0.6:
        hold_bonus = (1.1 - hold_time) * 0.6 + 1.2
elif hold_time >= 0.5:
        hold_bonus = 1.5
else:
        hold_bonus = hold_time + 1.0

raw_damage = weapon_damage * (clamp(hold_bonus, 1.0, 2.0) * 0.5 + 0.5)

if weapon_type == 'one_handed' or weapon_type == 'two_handed' or weapon_type == 'polearm':
        raw_damage *= math.pow(melee_damage_speed_power, speed_bonus)
elif weapon_type == 'crossbow' or weapon_type == 'bow' or weapon_type == 'throwing':
        raw_damage *= math.pow(missile_damage_speed_power, speed_bonus)
```

[569] Ikaguia, Modding Q&A.

[570] Caba'drin, Runtime Error! Help, please!.

[571] Caba'drin, Runtime Error! Help, please!.

[572] cmpxchg8b, [WB] Warband Script Enhancer v3.2.0.

[573] Formulas given by cmpxchg8b, Damage Formula and Damage Formula.

```

if weapon_type == 'crossbow':
    if raining:
        raw_damage *= 0.75
else:
    raw_damage *= proficiency * 0.01 * 0.15 + 0.85

if weapon_type == 'bow':
    raw_damage *= min(power_draw, difficulty + 4) * 0.14 + 1

    if mounted:
        raw_damage *= horse_archery * 0.019 + 0.8

    if raining:
        raw_damage *= 0.9
elif weapon_type == 'throwing':
    raw_damage *= power_throw * 0.1 + 1.0

    if mounted:
        raw_damage *= horse_archery * 0.019 + 0.8
elif weapon_type == 'one_handed' or weapon_type == 'two_handed' or weapon_type == 'polearm':
    raw_damage *= power_strike * 0.08 + 1.0

    raw_damage += strength / 5.0

    if (weapon_type == 'two_handed' or weapon_type == 'polearm') and (has_shield or mounted):
        raw_damage *= 0.85

        if weapon_type == 'polearm':
            raw_damage *= 0.85

        if weapon_flags & itp_two_handed:
            raw_damage *= 0.9

raw_damage = clamp(raw_damage, 0, 500)
```

```
armor = appropriate_armor_value_for_hit_location

if hit_shield_on_back:
armor += shield_resistance + 10
```

```

soak_factor = armor * module.ini_soak_factor_for_damage_type
reduction_factor = armor * module.ini_reduction_factor_for_damage_type

if item_flags & itp_extra_penetration:
soak_factor *= module.ini_extra_penetration_soak_factor
reduction_factor *= module.ini_extra_penetration_reduction_factor

randomized_soak = (random.random() * 0.55 + 0.45) * soak_factor
randomized_damage = (random.random() * 0.1 + 0.9) * raw_damage
soaked_damage = randomized_damage - randomized_soak

if (soaked_damage < 0.0):
soaked_damage = 0.0

randomized_reduction = math.exp((random.random() * 0.55 + 0.45) * reduction_factor
    * 0.014)
reduced_damage = (1.0 - 1.0 / randomized_reduction) * soaked_damage

if (reduction_factor < 0.00001):
reduced_damage = 0.0

damage_difference = round(reduced_damage + randomized_soak)
effective_damage = randomized_damage - damage_difference

if hit_bone == head:
effective_damage *= 1.2

if item_is_ranged:
effective_damage *= 1.75
elif hit_bone == calf or hit_bone == thigh:
effective_damage *= 0.9

effective_damage = clamp(effective_damage, 0.0, 500.0)
```

7.2.3 Weapon Speed Formula[574]

```
float mbAgent::getActionSpeed(int hand, bool reloading)
{
    mbItem item;
```

[574] Formula given by K700, Formula for the final weapon speed.

```
    item.m_itemKindNo = -1;

    if (reloading)
    {
        const mbItem *temp = getItem(hand);
        if (temp)
            item = *temp;
    }
    else
    {
        item = getWieldedItem(hand);
    }

    if (!item.isValid())
        return 1.0f;

    mbItemKind *itemKind = item.getItemKind();
    int type = itemKind->getType();
    mbItem secondaryItem = getWieldedItem(ah_secondary);
    float speed = itemKind->getSpeedRating() * 0.01f;

    if (itemKind->isWeapon())
    {
        float weaponFactor = type == itp_type_bow ? 0.11f : 0.07f;

        speed *= 0.01f * (int)getTroop()->getProficiency(itemKind->getProficiency(secondaryItem.isValid())) * weaponFactor + 1.0f - weaponFactor;
    }

    if (type == itp_type_two_handed || type == itp_type_polearm)
    {
        if (secondaryItem.isValid() || hasMount())
        {
            speed -= 0.15f;

            if (itemKind->m_properties & itp_two_handed)
                speed -= 0.05f;
        }
    }
    else if (type == itp_type_shield)
    {
        speed *= g_game->getTroopSkill(m_troopNo, skl_shield, true) * 0.03f + 1.0f;
    }

    if (g_basicGame.isMultiplayer())
        speed *= 0.03f * g_networkManager.m_combatSpeed + 0.94f;
    else if (itemKind->isMeleeWeapon() && (m_no != g_mission->m_playerAgentNo || rglConfig::Battle::iCombatSpeed > 2))
        speed *= 0.03f * rglConfig::Battle::iCombatSpeed + 0.94f;

    return speed;
}
```

7.2.4 Shot Speed/Missile Starting Velocity Formula

The formula is as follows:[575]

```
ingame_velocity_in_m_per_s = item_kinds_shoot_speed * sqrt((PD * 0.12) + 1.0) * 1.2
{PD capped at bow diff+4}
```

All missile speeds are (for unknown reasons) getting multipiled by 1.2, it does not matter which weapon type, except for missiles which are shot with the operation `add_missile`. Bow- and throwing shot speeds are also modified by power draw or power throw.[576]

The engine applies quadratic drag to missiles and the drag factors are defined in module.ini[577]:

```
air_friction_arrow = 0.002
air_friction_bullet = 0.002
```

7.2.5 Missile Speed Formula

The formula is as follows:[578]

```
float mbAgent::getMissileSpeed()
{
    float missileSpeed = 10.0f;
    float speedFactor = 1.0f;
    mbItem item = getWieldedItem(ah_primary);

```

[575] Formula posted by _Sebastian_, Modding Q&A, found at the Crpg forum.
[576] _Sebastian_, Modding Q&A.
[577] _Sebastian_, Modding Q&A.
[578] Formula posted by K700, Warband Script Enhancer 2 (v1.0.4.3).

```
    if (item.isValid())
    {
        mbItemKind *itemKind = item.getItemKind();

        missileSpeed = (float)itemKind->getMissileSpeed();

        if (itemKind->getType() == itp_type_bow)
        {
            speedFactor = rglMin(g_game->getTroopSkill(m_troopNo, skl_power_draw, true), item.getDifficulty() + 4) * 0.12f + 1.0f;

            if (m_horseAgentNo != -1)
                speedFactor *= g_game->getTroopSkill(m_troopNo, skl_horse_archery, true) * 0.019f + 0.8f;

            if (g_mission->m_weather.m_precipitationType == wpr_rain)
                speedFactor *= 0.9f;
        }
        else if (itemKind->getType() == itp_type_crossbow)
        {
            if (g_mission->m_weather.m_precipitationType == wpr_rain)
                speedFactor *= 0.75f;
        }
        else if (itemKind->getType() == itp_type_thrown)
        {
            speedFactor = g_game->getTroopSkill(m_troopNo, skl_power_throw, true) * 0.1f + 1.0f;

            if (m_horseAgentNo != -1)
                speedFactor *= g_game->getTroopSkill(m_troopNo, skl_horse_archery, true) * 0.019f + 0.8f;
        }
    }

    return rglSqrt(speedFactor) * missileSpeed * 1.2f;
}
```

7.2.6 Maximum Speed and Acceleration Formulas[579]

[579] cmpxchg8b, Modding Q&A.

```
maxSpeed *=1 ((agility * 0.7f + athletics * 3.0f + 25.0f) * 70.0f / (
    equippedItemsWeight + wieldedItemWeight * wieldedItemLength * 2.5 + 70.0f) +
    90.0f) / 100.0f
maxAcceleration = ((athletics / 6.0f + (agility + 2.0f) / 15.0f) * 40.0f / (
    equippedItemsWeight + 40.0f) + 1.0f) * 70.0f / (equippedItemsWeight + 70.0f) *
    5.0f
```

7.3 Other Remarks

Perhaps better split up latter

7.3.1 World Map Speed Formula

The game engine calculates the party speed with the following formula[580]:

```
float mbParty::getMovementSpeed()
{
    if (isShip())
    {
        return m_speedMultiplier * 15.0f;
    }
    else
    {
        float speed = m_speed * m_speedMultiplier;

        if (mbRegionTypeIsForest(m_regionType))
            speed *= 0.7f;

        if (g_game->isNight())
            speed *= 0.6f;

        return speed;
    }
}

float mbParty::calculateSpeed()
{
    float totalSpeed = 0.0f;
    float slowestTroopSpeed = 100000.0f;
    float totalWeight = ((m_flags & pf_carry_goods_mask) >> pf_carry_goods_shift)
        * 50.0f;
```

[580] Formula given by K700.

```
    int pathfinding = getSkill(skl_pathfinding);
    int numTroops = 0;
    int numRegulars = 0;
    int numPrisoners = 0;
    int numHorses = 0;
    mbParty tempParty;

    g_game->collectPartyAttachmentsToParty(m_no, tempParty);

    for (int i = 0; i < tempParty.m_numStacks; ++i)
    {
        mbPartyStack *stack = tempParty.getStack(i);
        mbTroop *troop = g_game->getTroop(stack->m_troopNo);
        float troopSpeed = troop->getMapSpeed();

        if (stack->isPrisoner())
            numPrisoners += stack->m_numTroops;
        else
            numRegulars += stack->m_numTroops;

        totalSpeed += stack->m_numTroops * troopSpeed;
        slowestTroopSpeed = rglMin(slowestTroopSpeed, troopSpeed);
        numTroops += stack->m_numTroops;

        if (troop->isHero())
        {
            int numInventorySlots = troop->getNumInventorySlots();

            for (int j = 0; j < numInventorySlots; ++j)
            {
                if (troop->m_inventory[j].isValid())
                {
                    mbItemKind *itemKind = troop->m_inventory[j].getItemKind();

                    if (itemKind->getType() == itp_type_goods)
                        totalWeight += itemKind->m_weight;
                    else if (itemKind->getType() == itp_type_horse)
                        numHorses++;
                }
            }
        }
    }

    float moraleFactor = (m_morale - 0.5f) * 0.05f + 1.0f;

    if (slowestTroopSpeed > 10000.0f)
        slowestTroopSpeed = 0.0f;

    if (totalSpeed == 0.0f)
        return 0.0f;

    float partyFactor = 1.0f / (totalWeight / (numTroops + 4 * numHorses + 3) * 0.007f + 1.0f) * (pathfinding * 0.03f + 1.0f) * 2.6f * moraleFactor;

    return (partyFactor * ((totalSpeed / numTroops) * 0.8f + slowestTroopSpeed * 0.2f)) / log((float)(numPrisoners / 2 + numRegulars + numHorses / 2 + 16));
}

void mbParty::updateSpeedMultiplier()
{
    g_basicGame.m_triggerResult = 100;
    g_game->executeMappedScript(game_get_party_speed_multiplier, m_no, 0);
    m_speedMultiplier = g_basicGame.m_triggerResult / 100.0f;
}
```

Working in here notes, also game script reference
world map travel speed[581]

7.3.2 Gain Experience Formula

When you gain experience the message “You got X experience.“ pops up. The formula is hard-coded and can thus not be edited by you, you can also not lose experience back.

```
(lvl+10)*(lvl+10)/5-1
```

You could however write a mission trigger firing when you kill an enemy to disable the original message, adding more experience on top of that, and finally reporting the total experience gained in a new custom message when it is enabled. This could look like as follows:[582]

```
(ti_on_agent_killed_or_wounded, 0, 0,[
    (store_trigger_param_1, ":victim"),
    (store_trigger_param_2, ":killer"),
```

[581] Dusk Voyager, Modding Q&A.
[582] Dusk Voyager, Modding Q&A.

```
    (store_trigger_param_3, ":is_wounded"),
    (agent_get_troop_id, ":victim_troop", ":victim"),
    (agent_get_troop_id, ":killer_troop", ":killer"),
    (try_begin),
        (eq, ":killer_troop", "trp_player"),
        (set_show_messages, 0),
        (store_character_level, ":victim_lvl", ":victim_troop"),
        (store_add, ":old_xp", ":victim_lvl", 10),
        (val_mul, ":old_xp", ":old_xp"),
        (val_div, ":old_xp", 5),
        (val_sub, ":old_xp", 1),
        (assign, ":new_xp", ":old_xp"),
        #[then increase new_xp any way you want as long as you are using val_xxx ops instead of store_xxx]
        (assign, reg3, ":old_xp"),
        (assign, reg2, ":new_xp"),
        (str_clear, s0),
        (str_clear, s1),
        (str_clear, s2),
        (str_store_troop_name, s0, ":victim_troop"),
        (str_store_troop_name, s1, ":killer_troop"),
        (try_begin),
            (eq, ":is_wounded", 0),
            (str_store_string, s2, "@killed"),
        (else_try),
            (eq, ":is_wounded", 1),
            (str_store_string, s2, "@knocked unconscious"),
        (try_end),
        (store_sub, ":diff", reg2, reg3),
        (add_xp_to_troop, ":diff", "trp_player"),
        (assign, "$player_has_recently_made_a_kill", 1),
    (try_end),
],
  [],
  ),
```

```
(0, 0, 0, [],
[(eq, "$player_has_recently_made_a_kill", 1),
(set_show_messages, 1),
(assign, "$player_has_recently_made_a_kill", 0),
(display_message, "@{s0} {s2} by {s1}", 0x008080), # a bit darker than the normal lovely teal text, but still
(display_message, "@You got {reg2} experience"),
]),
```

7.3.3 Experience Formula for Leveling Up

The game uses a fixed experience-required-to-level table, the required upgrade experience depends on the troop you upgrade from, not to. It has been generated with the following formula:[583]

```
int needed_upgrade_xp = 2 * (30 + 0.006f * level_boundaries[troops[troop_id].level
    + 3]);
```

However, the formula itself is not used by the game.[584]

```
 1) 0          14) 28832     27) 226879     40) 1721626     53) 21968215
 2) 600        15) 34362     28) 262533     41) 2070551     54) 27466219
 3) 1360       16) 40682     29) 303381     42) 2489361     55) 34338823
 4) 2296       17) 47892     30) 350164     43) 2992033     56) 42929679
 5) 3426       18) 56103     31) 412091     44) 3595340     57) 53668349
 6) 4768       19) 65441     32) 484440     45) 4319408     58) 67091786
 7) 6345       20) 77233     33) 568947     46) 5188389     59) 83871183
 8) 8179       21) 90809     34) 667638     47) 6231267     60) 160204600
 9) 10297      22) 106425    35) 782877     48) 7482821     61) 320304600
10) 13010      23) 124371    36) 917424     49) 8984785     62) 644046000
11) 16161      24) 144981    37) 1074494    50) 11236531    63) 2050460000
12) 19806      25) 168636    38) 1257843    51) 14051314
13) 24007      26) 195769    39) 1471851    52) 17569892
```

See the Leveling-related Parameters (Subchapter 6.2.3) entry fields at module.ini for the general settings which can be edited. For the implementation of extra rules (like if you want cavalry to require more XP, or bandits less, etc.) you need to edit the script `game_get_upgrade_xp`.[585]

7.3.4 Colour Codes

The format of colour codes in MABL follows the system 0xAARRGGBB. The first values stand for the alpha (transparency) component and the ones afterwards for the red, green and blue components, the classic RGB colour system converted to the HEX system. So to make RED you would use the code 0xFFFF0000 which would set red to 256, blue to 0 and green to 0. Knowing how to blend colours will help if you work with colours other than the standard ones.[586] Basic colour codes are as follows:

[583] Somebody, Modding Q&A.

[584] Answer given by cmpxchg8b, quoting a post of Lav, Modding Q&A

[585] kalarhan, Modding Q&A.

[586] jik, Updated Module System Tutorial, pages 18-19.

blue	0xFFAAAAFF	light blue	0xFFAAD8FF	red	0xFFFFAAAA
yellow	0xFFFFFFAA	pink	0xFFFFAAFF	purple	0xFF6AAA89
black	0xFFAAAAAA	white	0xFFFFFFFF	green	0xFFAAFFAA
brown	0xFF7A4800				

The alpha takes as well values within the range of 0 and 255 in hex system, where 0 (0x00) is full transparent (invisible) and 255 (0xFF) is not transparent at all.[587]

Calculating colour codes via code[588]

7.3.5 Usage of Unicode Characters

Work in the notes here.[589]

ASCII characters into Warband MP[590]

Scandinavian characters[591]

extra letters (unicode ASCI characters)[592]

7.3.6 Weapon Switch System

You might want to change the way a character switches from a weapon to another, for example such way that the first equipped weapon should be put away automatically with an animation and the character draws afterwards the other weapon. The behaviour in Warband is however hard-coded. The only way might be to put something together by playing animations in the ti_on_item_wielded and ti_on_item_unwielded triggers.[593]

7.3.7 Calculation for Reticle Size at Aiming with Ranged Weapons

The reticle size is getting determined by the accuracy and the damage value of the ranged weapon. You can reduce the reticle size by raising the accuracy and by lowering the damage and vice versa.[594]

7.3.8 Kicking

A kick is an area-of-effect (AOE) attack for which the game engine checks for the agent in front of the kicker.[595] You cannot change the range of a kick, it is hard-coded into the game.[596]

[587] Dalion, Mount & Blade Modding Discord.

[588] Somebody, Modding Q&A, and dunde, Python Script/Scheme Exchange.

[589] Swyter, Modding Q&A.

[590] Hm, Modding Q&A.

[591] SupaNinjaMan and MadocComadrin, Modding Q&A.

[592] kalarhan, Modding Q&A.

[593] cmpxchg8b, Modding Q&A.

[594] _Sebastian_, Modding Q&A.

[595] cmpxchg8b, [WB] Warband Script Enhancer v3.2.0.

[596] The_dragon, Modding Q&A.

7.3.9 Chance for horses being crippled or killed

When horses “die“ while being ridden by the player in battle there is a random chance for it to be crippled or killed after the battle. You cannot increase or decrease the chance for that, it is hard-coded into the game engine. You can still write another script that does a similar thing by checking and replacing the horse’s item modifiers after a mission and create thus a workaround for it.[597]

7.3.10 Crosshair

The crosshair can be found at the `ui_gadgets.dd` texture, the part of the texture which contains the Native crosshairs is in the upper left section. If you are only getting a color change it might be because you have not edited the alpha map - this is the part of the texture which will define the ßhapeöf your crosshair - alpha textures define what is visible and what is not.[598]

7.3.11 Ammo Drop and Pick Up

When you drop a bow, crossbow or firearm, you also drop the ammunition. This is a hard-coded behaviour of the game engine. However, you can give the agent a new bag of ammunition when the item is dropped via the usage of triggers. An appearing problem might then be that the player can just pick up the old ammo again unless you give it a flag.[599]

It is also a hard-coded behaviour of the game engine that a picked up arrow (or other ammo) will first fill up the ammunition (of the same kind) a player is already carrying around. A picked up fire arrow might thus become a simply arrow unless the ammunition is filled up to its capacity and the fire arrow is getting assigned to another weapon slot (if a slot is free that is).

7.3.12 Melee Weapon Bounce

When striking an enemy the player might experience that the melee weapon bounces and the thrust gets repulsed. There are multiple factors which can influence this, so the following might be an incomplete list:[600]

1. The “business end“ of a weapon is either a line or a long capsule running down the weapon’s length, which deals damage if it collides with an agent hitbox.

2. Weapons bounce if:

[597] Somebody, Modding Q&A.
[598] La Grandmaster, Modding Q&A.
[599] Somebody, Modding Q&A.
[600] jacobhinds, Modding Q&A.

   (a) the velocity of collision is too low

   (b) the weapon's "business end" hits the hitbox at an odd angle relative to the attacker, like spinstabbing

   (c) the weapon doesn't do enough damage

   (d) the "shaft" hits the hitbox (how the game determines shafts is unknown)

3. Weapons also bounce depending on weapon type and length. There seems to be a lot more lenience to short weapons regarding the "shaft" but at the same time polearms seem to get less lenience than weapons of the same length defined as swords. This is just an observance though, it is untested.

4. All this is based on the animations. This allows custom attack animations to affect the combat model.

7.3.13 Biggest Value that a Variable can hold

Work in comment here[601]

7.3.14 Semi-randomness

Some stuff about semi-randomnes[602]

7.3.15 Save Game Compability

What are changes which will break the save game compabilities?

1. **New Global Variables, or changes in global variables order.** How to deal with it: Before modifying the Module System, compile it once again. Copy variables.txt from the Module folder to the Module System's folder. Any new introduced global variables must be included into the variables.txt file by overwritting reversed_## global variables.

2. **New Simple Triggers.** How to deal with it: Don't create any new simple triggers. If you badly need new simple triggers, just modify

   ```
   (24,
   []),
   ```

   on the last parts of module_simple_triggers.py

[601] dunde, Modding Q&A.

[602] MadocComadrin and Caba'drin, Modding Q&A.

3. **Inserted new items, new factions, new troops, etc. especially changing constants like mercenaries_end, books_begin, etc.** How to deal with it: Don't do it.

Work in the notes of the threads here[603]

7.3.16 Quoted and Unquoted Variables

Work in those notes here[604]

7.3.17 Uninitiated Variables

Variables in the Module System don't need to be created with the `assign` operation. They share however their memory address and can show a weired behaviour in some cases (like fast loops) or OS (like Linux and Mac).[605]

The game engine will overflow uninitiated variables at the operation systems Linux and Macs. If you ever saw a budget report with trillions of coins of deficit, that is why. In most cases the variables get thus assigned the values -1 or 0 in the Module System, even if something else is going to be stored in them a few lines later. Global variables are different from local variables, as they are stored on the savegame (as slots) and thus always have a defined value (including a default initial value).

The engine does also not handle loops well. So it is not uncommon to see cases of ghost values causing unpredictable bugs. So for those cases your best bet is to make sure to not depend on defaults. That is mostly for local variables (which use the same register in memory).

7.3.18 Mount&Blade Engine and Multithreading

interesting discussion about multithreading[606]

7.3.19 List of Native Key Kombinations

pausing a scene (listing available button combos)[607] Rear horse, ctrl + j

7.3.20 In-game Cursor Info

The cursor is a part of the window itself, and so you cannot use shaders to alter it.[608]

[603] NPC99, Modding Q&A, kalarhan, Modding Q&A, Modding Q&A and Modding Q&A.

[604] Caba'drin, Modding Q&A, and Vornne, Modding Q&A.

[605] kalarhan, Modding Q&A, Modding Q&A and Modding Q&A.

[606] _Sebastian_, Modding Q&A.

[607] Docm30, Modding Q&A.

[608] dstn, Mount & Blade Modding Discord.

8 Most Common Errors at Modding

Chapter here about how to read the error and warning messages at compiling, somewhere is also a link around of kalarhan's note in VC Q&A[609]

8.1 Most Common Errors at Compiling[610]

If not everything did go well, check carefully for spelling and syntax. Make very sure that all commas and brackets are in the right place. Bad syntax is the most common source of compiler errors in the official module system. After compiling, errors are usually pointed out at specific lines. Using a code editor like NOTEPAD++ makes it easier as lines are numbered, and it highlights (when you mouse over one) both the beginning and ending brackets.

Since I figured it would be useful, I've gone ahead and compiled a little database containing the most common errors you're likely to encounter when building the Python code into a working mod, and their meanings.

```
SyntaxError: invalid syntax
```

Meaning: Missing a bracket, comma or quotation mark – [], (), , or – OR you have too many of them.

```
TypeError: 'tuple' object is not callable
```

Meaning: Missing a comma just before an open bracket (

```
TypeError: list indices must be integers
```

Meaning: Missing a comma/too many brackets.

```
ERROR: INPUT TOKEN NOT FOUND: <name>
NameError: global name 'cause_error' is not defined
```

Meaning: You have a starting dialog-state – example, `[trp_guard,"guard_introduce_1",` – that isn't being led to by an ending dialog-state – example, `"close_window",[]],`

[609] kalarhan, Modding Q&A.

[610] Taken from the *Modding Q&A* [*For Quick Questions and Answers*] by Winter with additions from *Updated Module System Tutorial* by jik and from *Common Errors and How to Fix Them* by Yoshiboy.

```
ERROR: INPUT TOKEN NOT FOUND: <empty>
NameError: global name 'cause_error' is not defined
```

Meaning: You have an empty starting dialog-state – example, [trp_guard,"guard_introduce_1",

```
NameError: name '<something>' is not defined
```

Meaning: Missing a prefix such as trp_, itm_ or mnu_ OR missing a name OR misspelled a name.

```
IndexError: tuple index out of range
```

Meaning: Nebulous. If you're modding module_mission_templates.py, look for undefined alter flags and undefined AI flags, these are the most common source.

```
TypeError: int argument required
```

Meaning: Missing an integer such as a mission-template flag. You can find out which calls are integers by looking at Armagan's documentation at the beginning of each module file, (int) calls being integers.

```
IndexError: list index out of range
```

Meaning: There's an error in one of your list calls – a list is anything between brackets.

```
Error: Unable to find object:<name>
ERROR: Illegal Identifier:<name>
```

Meaning: The object you're trying to call does not yet exist OR there is a misspelling in the name. It could also be that you forgot to put a $ or : on the variable.

```
WARNING: Local variable never used:<name>
```

Meaning: It means exactly what it says: A local variable is never used. Say you have a local variable named ":cur_day". If you declare ":cur_day" but you never ever use it in the script, it will be then a local variable which is not used.

```
SCRIPT WARNING ON OPCODE -<negative value>:
```

Meaning: A negative OPCODE means that you are using a combo like `this_or_next|SOMETHING` or `neg|check_quest_active`, etc., so find that line of code and see what you did wrong. The error tells you the trigger number, line of code, and what is wrong (invalid parameter)[611]

Not sure if this string compiler error is still present, would need to test for it.

multiple level ups at game start[612] leveling at game start problem[613]

Referencing .py files

Make sure that the files at which you use implemented constants are referencing to the respective module_*.py or headers_*.py file at which they have been declared. Alternatively you need to use directly the number of the constant, reading them out of the .py file in which they are declared. You reference .py files to each other via a line like for example from module_constants import * at the top of the file. Check at the top of each module_*.py file which other ones are getting referenced already. Keep in mind that you mostly need to reference the header files since the usage of the Module System Prefixes mentioned in Subsubchapter 4.1.2 do the work for you at the other module_*.py files already.

If you want to raise the attribute of a specific troop for example and do it in module_scripts.py. Intuitively you would try to use this line:

```
(troop_raise_attribute,"trp_player",ca_strength,-2),
```

Compiling with it would however result in some error message since the compiler does not know what `ca_strength` stands for. Those are constants declared in header_troops.py. The situation is thus similar to the one at which you make use of a constant or variable which has not been declared yet. So you either need to use directly the number of the respective attribute (where 1=STR, 2=AGI, 3=INT and 4=CHA) as follows:

```
(troop_raise_attribute,"trp_player",1,-2),
```

or you need to add a reference line at the top of module_scripts.py as follows:

```
from header_troops import *
```

8.2 Most Common other Errors

```
map_scene->manifold->faces[target_manifold_index].tag != rt_water (or rt_mountain)
```

[611] kalarhan, Modding Q&A.
[612] Lumos, Modding Q&A and Modding Q&A.
[613] Lumos, Modding Q&A.

Meaning: You have a party on the world map that is trying to spawn on water or mountain terrain. Check all your spawn points and spawn radius'.

```
Get Object failed for *
```

Meaning: Any get object failed error is where in one of your BRFs or items/scenes/scene props in-game you are referencing a texture, material or mesh that does not exist. Check the spelling and that everything is referenced properly, remember that capital letters count! Check if your BRFs are loading in the right order.

```
Buffer.has_more_tokens()
```

Meaning: The message means there is a mismatch between the expected and actual size of a data structure. This could be caused by a malformed BRF, a corrupted .txt file, or a savegame incompatibility. It might also be caused by reading past the end of a data structure, e.g. you only have 25 strings and you ask for string #57. It can also be when you use a space instead of a _ in identifier names. It is a complicated error which can be caused by a variety of things.

```
rgl_between(skill_level,0,(skills[skill_no].max.level+1))
```

Meaning: This assert is checked when the game reads your skills. It verifies that the number read was not more than the possible maximum, no matter the dependent attribute (Int, Str, Dex, Cha). Typically, it means that you are either reading a skill from a nonexistent troop or reading a nonexistent skill, or both. This error can sneak up on you when you are making troops using predefined skill sets like `knows_common` and add additionally more skill points which let a skill exceed the maximum value.

```
rgl_between(current_sentence, 0, num_sentences)
```

Meaning: Caused when talking to someone who has nothing to say.

Non ASCII charaters, reproduce error message.[614]
error on last line[615]

8.3 World Map Issues

Create an own world map chapter

[614] Antonis, Mount & Blade Modding Discord.
[615] jacobhinds, Modding Q&A.

World map can behave strange if separated by rivers and fords into non-connected parts.[616] And again.[617]

Not really fitting here but collecting world map info snippets here until new subchapter is opened for it.

One can't define new ground textures, but one should be able to modify flags, meshes, colors etc of existing ones.[618], replace the existing ones[619]

Training fields[620]

Deep sea effect (find out more about it)[621]

largest possible world map[622]

campaign map speed[623]

World map stuff[624]

Limit of vertices per mesh[625]

Some strings getting used all the time?[626]

hard-coded map materials[627]

Mixture for creating a campaign map[628]

Hard-coded material for trees[629]

worldmap questions[630]

Deep ocean material? Also swamp discussion, check upfollowing page too[631]

ford texture material[632]

world map inland sea maelstroem[633]

Deep ocean[634]

ocean world map shader[635]

[616] GetAssista, Modding Q&A.
[617] Lumos, Modding Q&A.
[618] Somebody, Modding Q&A.
[619] Dargor, Modding Q&A.
[620] Ritter Dummbatz, Modding Q&A.
[621] Cozur, Modding Q&A.
[622] Discussion.
[623] MadVader, Modding Q&A.
[624] Lav, Modding Q&A.
[625] cmpxchg8b, Global Map Visualiser.
[626] Somebody, Modding Q&A.
[627] Swyter, Modding Q&A and Modding Q&A.
[628] jacobhinds, Modding Q&A.
[629] jacobhinds, Modding Q&A.
[630] jacobhinds, Modding Q&A.
[631] jacobhinds, Lav, La Grandmaster, Seek n Destroy and Dusk Voyager,Modding Q&A.
[632] Michadr and produno (credit), Modding Q&A.
[633] Discussions, Modding Q&A and Modding Q&A.
[634] jacobhinds, Modding Q&A.
[635] La Grandmaster, Modding Q&A.

worldmap forest performance[636]

Thorgrim map editor issue[637]

hard-coded world map terrain[638]

random terrain forest problem[639]

transfer changes from map editor to ms[640]

If own chapter, world map speed (also with K700 formula)[641]

campaign map crashes[642]

unanswered mountain shifts camera up[643]

s1 worldmap string[644]

world map textures[645]

world map icons limit (crossreference to map icons)[646]

World map with seasons[647]

some sailing stuff[648]

ground specs[649]

shore effect[650]

Shiny water at world map and flatter terrain[651]

world map landmasses[652]

8.4 Errors appearing in-game

Insane level up at game start[653]

CTD with error message : Hash Vector failed at index -100000.[654]

hard-coded stuff not getting referenced in official MS[655]

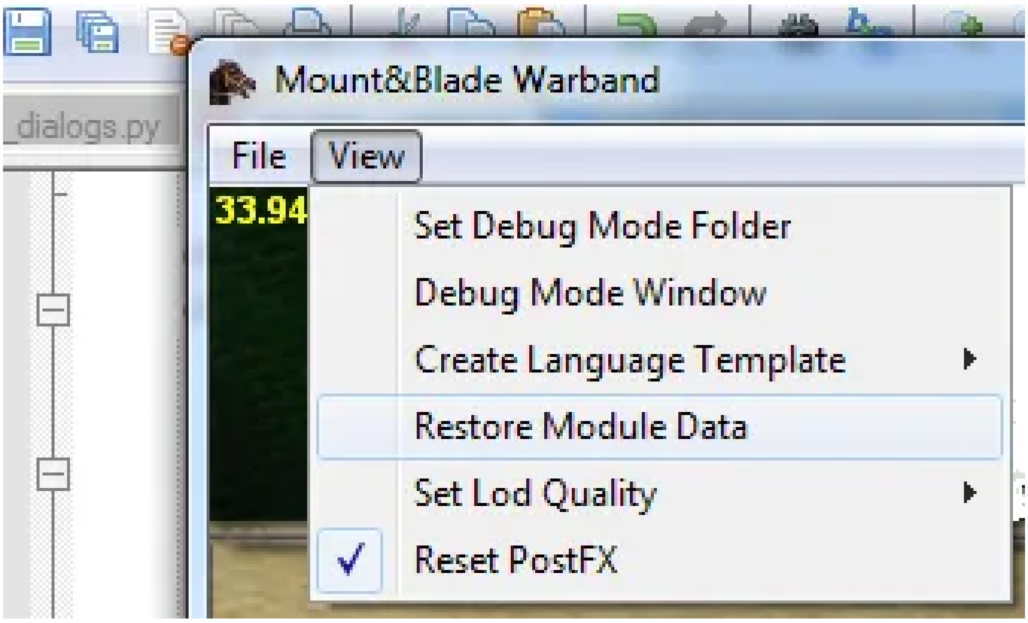
[636] cmpxchg8b, Modding Q&A.
[637] MadVader, Modding Q&A.
[638] The Bowman (credit), Modding Q&A.
[639] The Bowman, Modding Q&A.
[640] NPC99, Modding Q&A.
[641] kalarhan, Modding Q&A.
[642] NPC99, Modding Q&A.
[643] NPC, Modding Q&A.
[644] kalarhan, Modding Q&A.
[645] NPC99, Modding Q&A.
[646] NPC99, Modding Q&A.
[647] Docm30, Modding Q&A.
[648] NPC99, Modding Q&A.
[649] NPC99, Modding Q&A.
[650] NPC99, Modding Q&A.
[651] Leprechaun (credit) and Hellequin, Useful Techniques
[652] cmpxchg8b, [WB] Warband Script Enhancer v3.2.0.
[653] Lumos, Modding Q&A.
[654] cmpxchg8b, Modding Q&A.
[655] The_dragon, Modding Q&A.

9 Good Practice

Link Constantly back up your files!

Secondary compilation for backup[656]

Press Ctrl+M for reloading all txt files anew.[657] Perhaps not working anymore, I think it does on Wine?[658] Remark of myself for windowed mode reload[659]



useful tips at modding[660]

Modding tip[661]

Recommendations[662]

engine limits and potential for optimisation[663]

ideal order for total conversion mods, also next page[664]

better more little scripts? Read rest of discussion too[665]

read compiler messages[666]

how to use rgl log to debug the game[667]

[656] MadocComadrin, Modding Q&A.
[657] GetAssista, Modding Q&A.
[658] Caba‘drin, Modding Q&A, and Highlander, Modding Q&A.
[659] Earendil, Modding Q&A.
[660] grailknighthero (credit), Modding Q&A, and The_dragon, Modding Q&A.
[661] Lav, Modding Q&A.
[662] Lumos, Modding Q&A.
[663] Lumos, Modding Q&A.
[664] Multiple participants, Modding Q&A.
[665] kalarhan, Modding Q&A.
[666] kalarhan, Modding Q&A.
[667] kalarhan, Modding VC: basic tutorials and Q&A thread.

system impact of triggers, scripts, etc[668]

unassigned variables break mac/linux? Also battle size and other stuff[669]

coding tip[670]

Modding philosophy[671]

modding practise, mark own changes[672]

Modding philosophy (kalarhan quoting himself) [673]

compiles fine does not mean there are no bugs[674]

https://9gag.com/gag/aB2KB9N and https://9gag.com/gag/ad8NEqZ, image research and comments.

game stuttering, optimization[675]

differentiating things, modding approach[676]

Some other compiling faults[677]

features can collide with each other[678]

Reading compiler errors[679]

Es handelt sich hierbei um ein fehlendes Komma oder eine fehlende Klammer. Beachte - Wie immer ist der Fehler über der vom Compiler angegeben Linie zu suchen.

built-in script debugger[680]

9.1 Multiplayer and hard-coded presentations related notes

Always test multiplayer mods with a dedicated server.[681]

Maximum server events[682]

Setup a mp server[683]

Multiplayer gold presentation (perhaps integrate later at chapter Hard-coded presentations)[684]

[668] jacobhinds and kalarhan, Modding Q&A.
[669] kalarhan, Modding Q&A.
[670] jacobhinds and kalarhan, Modding Q&A.
[671] kalarhan, Modding Q&A and Modding Q&A.
[672] EmielRegis (credit), Modding Q&A.
[673] kalarhan, Modding Q&A.
[674] kalarhan, Modding Q&A.
[675] _Sebastian_, Modding Q&A.
[676] kalarhan, Modding Q&A.
[677] Earendil and kalarhan, Modding Q&A.
[678] kalarhan, Modding Q&A.
[679] Dalion, Modding Q&A.
[680] cmpxchg8b, [WB] Warband Script Enhancer v3.2.0.
[681] Vornne, Modding Q&A
[682] cmpxchg8b, Modding Q&A.
[683] Vornne, Modding Q&A.
[684] Caba‘drin, Modding Q&A, and Caba‘drin,Modding Q&A, and Caba‘drin, Modding Q&A.

Client-side trigger intervals are timed separate for each client?[685]

ti_on_agent_hit is not client side?[686]

Subevents explained a bit[687]

server side vs client side[688]

Adding an mp event[689]

Spectator code is entirely hard coded.[690]

Team score display[691]

prevent host MP function[692]

Some useful info snippets[693]

Gun sound (Yoshiboy mp tutorial not any longer useful for sounds but otherwise good)[694] Sound file format seems to be important (jump a bit messages after this one)[695]

Horses are in team 7 at MP[696]

Change cloth in MP character window[697]

Maximum of 250 Players at server (200 hard-coded?)[698]

Non-server players can be any number between 1 and max players[699]

MP spawn as different race (and issues)[700]

MP weather[701]

Icon above head hard-coded?[702] custom presentation[703]

Some hard-coded presentation MP stuff[704]

Accessible MP stuff (hard-coded presentations)[705]

agent_set/get_group for multiplayer[706]

[685] Theoris, Modding Q&A.

[686] cmpxchg8b, Modding Q&A.

[687] rabican, Modding Q&A.

[688] Vornne, Modding Q&A.

[689] Vornne, Modding Q&A.

[690] Vornne, Modding Q&A.

[691] Caba‘drin, Modding Q&A.

[692] dunde, Modding Q&A.

[693] Vornne, Modding Q&A.

[694] Specialist, Modding Q&A.

[695] Specialist, Modding Q&A.

[696] Mammoet, Modding Q&A.

[697] There is also something about tableau to do, Modding Q&A.

[698] cmpxchg8b, Modding Q&A.

[699] MadocComadrin, Modding Q&A.

[700] Meneldur, Modding Q&A, Meneldur, Modding Q&A, and Meneldur, Modding Q&A.

[701] Caba‘drin, Modding Q&A.

[702] captain lust, Modding Q&A and Modding Q&A.

[703] Somebody, Modding Q&A.

[704] Caba‘drin, Modding Q&A.

[705] Caba‘drin, Modding Q&A.

[706] SonKidd, Modding Q&A.

loading bar and loading screen text.[707]

Main menu presentation.[708]

Hard-coded cinematic intro.[709]

maximum mp troops 200 (down from 250)[710]

Player banner[711]

customize the kill image for specific weapons[712]

server and client events[713]

never test scripts with 'host a game' in mp[714]

Module version display[715]

mp spectator camera hard-coded[716]

launcher picture[717]

3D character conversation window[718]

mp give player certain facecode[719]

commanding flag hard-coded[720]

teleport player mp synchro[721]

some start menu stuff, covered buttons can still get pressed[722]

hard-coded presentation for start menu[723]

Hard-coded mp UI strings[724]

time of day string[725]

random main menu[726]

daytime mp[727]

[707] Caba'drin, Modding Q&A.
[708] Caba'drin and Somebody, Modding Q&A.
[709] Specialist, Modding Q&A.
[710] Modding Q&A.
[711] Ikaguia (credits?), Modding Q&A.
[712] cmpxchg8b, Modding Q&A.
[713] shokkueibu, Modding Q&A.
[714] Vornne, Modding Q&A and Modding Q&A.
[715] cmpxchg8b, Modding Q&A.
[716] Vornne, Modding Q&A.
[717] Modding Q&A.
[718] MadVader, Modding Q&A.
[719] Modding Q&A.
[720] Somebody, Modding Q&A.
[721] The_dragon, Modding Q&A and Modding Q&A.
[722] Lumos, Modding Q&A.
[723] MadVader, Modding Q&A.
[724] Somebody, Modding Q&A.
[725] The_dragon, Modding Q&A.
[726] dunde, Modding Q&A.
[727] The_dragon, Modding Q&A.

opening video[728]

loading screen stretch[729]

text highlight colours menu (check linked content if present already)[730]

multiplayer menu[731]

hard-coded conversation window[732]

Main menu presentation[733]

Disable hard-coded order panel[734]

clicking on map buttons[735]

battle log in mp[736]

bot formations mp[737]

hard-coded player face code and f1 flag[738]

hard-coded dialog window stuff[739]

random music at main menu (at bottom of post)[740]

start menu[741]

hard-coded order menu[742]

background campaign map buttons[743]

character screen[744]

ui buttons hook[745]

hard-coded presentations buttons named wrongly[746]

Bots in multiplayer no variety[747]

play sound stuff mp[748]

[728] Lumos, Modding Q&A.

[729] Modding Q&A.

[730] Swyter, Modding Q&A.

[731] Seek n Destroy, Modding Q&A.

[732] Swyter, Modding Q&A.

[733] cmpxchg8b, Modding Q&A.

[734] Swyter, Modding Q&A, and Somebody, Modding Q&A.

[735] Somebody, Modding Q&A.

[736] Lav, Modding Q&A.

[737] Lav, Modding Q&A.

[738] jacobhinds, Modding Q&A.

[739] Lumos, Modding Q&A.

[740] kalarhan, Modding Q&A.

[741] kalarhan, Modding Q&A.

[742] _Sebastian_, Modding Q&A.

[743] litdum (credit), Modding Q&A.

[744] kalarhan, Modding Q&A.

[745] kalarhan, Modding Q&A.

[746] TortenSkjold, Mount & Blade Modding Discord.

[747] Somebody, Modding Q&A.

[748] Vornne, Modding Q&A.

mp sound[749] 20m radius for that operation[750]

mp maximum player 200[751] Hack server to increase limit[752]

item categories mp[753]

lock an agents rotation MP[754]

Slot maximum mp and server events limit[755]

mp find spawn point[756]

custom kill info[757]

controll agent mp problem[758]

get rid of unwanted agent in mp[759]

spawning mp[760]

spawning player at specific entrypoint in mp[761]

some mp notes[762]

mp spectator and unassigned[763]

Difference SP and MP team[764]

Player slots are used in MP, but they are effectively the same things as Agent slots.

mp player equipment[765]

mp phyics[766]

mp differentiate between oldies and newcomers at map change[767]

MP troop limit (still active? I think yes, recheck)[768]

hard-coded order menu, PBOD workaround[769]

mp additional teams for bots[770]

[749] _Sebastian_, Modding Q&A.

[750] The_dragon, _Sebastian_ and cmpxchg8b, Modding Q&A and Modding Q&A.

[751] Lumos, Modding Q&A.

[752] cmpxchg8b, Modding Q&A.

[753] Somebody, Modding Q&A.

[754] Somebody, Modding Q&A.

[755] Vornne, Modding Q&A.

[756] Dusk Voyager, Modding Q&A.

[757] Somebody, Modding Q&A.

[758] WookieWarlord (credit), Modding Q&A and Modding Q&A.

[759] WookieWarlord, Modding Q&A.

[760] The_dragon, Modding Q&A.

[761] The_dragon, Modding Q&A.

[762] The_dragon, Modding Q&A, and Arch3r, Modding Q&A.

[763] Namakan (credit), Modding Q&A.

[764] Vornne, Modding Q&A, and InVain, Mount & Blade Modding Discord.

[765] Vornne, Modding Q&A.

[766] NPC99 redirects, Modding Q&A.

[767] _Sebastian_, Modding Q&A.

[768] Somebody, Modding Q&A.

[769] Efe Karacar (credit), Modding Q&A.

[770] Vornne, Modding Q&A.

A bit at multiplayer send int to server [771]

game receive url response[772]

reset map/round[773]

killed a teammate message[774]

hard-coded mp chat[775]

mp equipment multiple rows[776]

override items mp[777]

spawn a player in mp[778]

edit hard-coded menu text colour[779]

conversation window[780]

MP weapon icon also influenced by damage type [781]

Remove/replace round icons above head[782]

Warband dedicated server three threads[783]

inventory stat important for mp troops?[784]

triggers and simple triggers no effect in mp[785]

10 In-game Editor

The in-game editor allows decoration of scenes with items and new buildings (commonly called scene props), editing of scene settings and informations, generation of terrain for use in scenes, editing of NPC faces.

Some remarks to add if not done at other places already.[786]

[771] Vornne, Modding Q&A.

[772] Swyter, Modding Q&A.

[773] Pitch, Modding Q&A.

[774] kalarhan, Modding Q&A.

[775] Pitch, Namakan and Maroon, Modding Q&A.

[776] Somebody, Modding Q&A.

[777] Somebody, Modding Q&A.

[778] AlbertFaubrein, How to spawn the player character in multiplayer?.

[779] Earendil, Modding Q&A.

[780] Multiple participants, Editing conversation window.

[781] Seek & Destroy, Mount & Blade Modding Discord.

[782] Somebody, Warband Script Enhancer v3.2.0.

[783] cmpxchg8b, Warband Script Enhancer v3.2.0.

[784] Somebody, Modding Q&A.

[785] The_dragon, Modding Q&A.

[786] _Sebastian_ and Caba‘drin, Modding Q&A.

10.1 Using the Edit Mode for Scening

10.1.1 How to get to Edit Mode

Need to update this subsection a bit. Taken for now from Ursca.

First, start M&B, and go to 'configure' at the beginning. In the last tab (Advanced), click the tick-box for 'edit mode'. To set up a blank slate for your scene, go to your M&B folder > Modules. Copy the native folder, paste it and rename it 'Mod scenes' or similar. Start M&B again, choose the 'Mod scenes' module from the menu and play. Make a quick character and go to Zendar. Now press Alt+Enter to go into windowed mode. Press Ctrl+E to open the Edit Mode window.

10.1.2 Controls

- Click and drag - look around
- Right click - Select object
- W - forward
- A - left
- S - back
- D - right
- F - Up
- C - down
- Ctrl + any movement key - go fast!
- Delete - Delete object

Start off by clearing the Zendar objects out of the scene by right clicking them and pressing delete. Do this to everything apart from the the yellow arrows.

10.1.3 Adding an Object

In the edit mode window (Windowed - Alt+Enter) go to 'Edit Objects', in the bottom window (Add object) scroll through the list to find an object you would like to place in the scene. Click the name and click add object.

To move the object, hold 'G' and move to mouse to move it on a horizontal plane. Hold 'T' and move to move it vertically. Holding Z,X or Y and dragging will rotate the object around the respective axis.

To deselect the object, right click off it. If you want to select an object to move, right click it.

10.1.4 Painting the Ground Texture

Click 'ground paint'. choose a ground texture from the drop-down menu. Radius changes the size of the 'brush'. Weight changes the opacity of the brush. Hardness changes the softness of the edges of the brush.

10.1.5 Spawnpoints in Native

Old (outdated) list with spawnpoints in Native by Amman d Stazia

entry point reference for siege scenes (SP)[787]

custom battle scenes spawn points[788]

entry points random terrain[789]

entry points regular battles[790]

10.1.6 Other notes

AI mesh game crashing[791]

Upper limit of scene props at scene.[792]

hard-coded terrain vertex colours[793]

blue underwater fog (?)[794]

random entry points if not placed[795]

underground building in scene, shooting not possible underground[796]

ai meshes siege tower [797]

terrain mountains[798]

[787] MadVader, Modding Q&A.

[788] Docm30, Modding Q&A.

[789] Lumos, Modding Q&A.

[790] cwr, Modding Q&A.

[791] Gergin Adam (Credit), Modding Q&A.

[792] Docm30, Modding Q&A.

[793] Somebody, Modding Q&A.

[794] Dusk Voyager, Modding Q&A.

[795] Somebody, Modding Q&A and Modding Q&A.

[796] jacobhinds, Modding Q&A, and MadocComadrin, Modding Q&A.

[797] Husselism, Modding Q&A.

[798] jacobhinds, Modding Q&A.

worldmap forest performance[799]

wheat and general scene prop count[800]

ai intermediate points[801]

vertices limit at scene[802]

Raining outside but not in interior[803]

Ai decission making at choosing a way[804]

Ai limiter[805]

blue plane edit mode[806]

reset terrain at scene editing[807]

first generate terrain, then edit scene[808]

water_river[809]

mirror scene props[810]

vertex colour on scenes[811]

Pathfinding AI mesh[812]

10.2 Ingame Troop Viewer

There is an ingame troop viewer. To use it Check Edit Mode, start a game and go to the character screen - there are new arrow buttons there that let you view all the troops.[813]

[799] jacobhinds and cmpxchg8b, Modding Q&A.
[800] jacobhinds, Modding Q&A.
[801] cwr, Modding Q&A.
[802] jacobhinds, Modding Q&A.
[803] Vornne, Nordous' Sceners Guild.
[804] Ruthven, Modding Q&A.
[805] EmielRegis (credit), Modding Q&A.
[806] Ruthven, Modding Q&A.
[807] InVain, Modding Q&A and Modding Q&A.
[808] kalarhan, Modding Q&A.
[809] Kekse (credit), Modding *Sammelthread*.
[810] Kekse (credit), Modding *Sammelthread*.
[811] Various, [WB] Warband Script Enhancer v3.2.0 and [WB] Warband Script Enhancer v3.2.0.
[812] MadocComadrin, Modding Q&A.
[813] MadVader, Modding Q&A.

11 Frequently Asked Questions

Placeholder section for stuff for which I have no idea yet where to integrate it otherwise

11.1 What is a tuple?

A tuple is a set of data fields surrounded by normal brackets. Every operation or structure definition in Module System is a tuple. Take module_scripts.py for example:

```
scripts = [
  ("script_name",
    [
      (operation, ":param1", ":param2"),
      (another_operation, ":param1", ":param2", ":param3"),
    ]
  ),
  ("another_script_name",
    [
      (operation, ":param1", ":param2"),
      (another_operation, ":param1", ":param2", ":param3"),
    ]
  ),
]
```

What we have here is the array (square brackets, "scripts=[]") which contains two tuples (surrounded by normal brackets). Each of those tuples is a script definition. Each script in turn contains one string value (name of the script) and an array of operations. Each operation is once again a tuple, with first data value being the operation name, and the rest are operation parameters. So we have tuples inside array inside tuple inside another array. Simple eh? Especially when compared to more complex files like module_presentations.py.[814]

11.2 What is the difference between `neg` and `neq`?

`neg` is for the opposite/inverse of the operation whie `neq` stands for "´not equal“.

11.3 What are Fixed Point Values/Numbers?

Short explanation here.

[814] Lav, Modding Q&A.

11.4 Why can't I see the changes I made to my mod?

Many things are stored in the save files, and so changes made to your mod may only be visible when you start a new game. Not everything requires you to start a new game however.[815] Need to expand explanation here.

11.5 How to read OPCode errors?

Work in that part here.[816]

11.6 Is there an upper limit to the number of troop types or cities/villages/castles available?

The only limit is your imagination (and RAM).[817]

11.7 How to create a Native Compatible Multiplayer Mod?

Work in notes here.

limitations server side native mod[818]

server side mod[819]

item selection client side[820]

server side throwing weapon[821]

changing order causes problems[822]

11.8 Why is the initial loading time of a new game getting longer?

Work in note here.[823]

11.9 How does the Auto-Calc Battle work?

auto-calc battle[824] Rabbit hole threads[825]

[815] Thorgrim, Modding Q&A.
[816] Caba'drin, Modding Q&A.
[817] Somebody, Modding Q&A.
[818] The_dragon, Modding Q&A.
[819] The_dragon, Modding Q&A.
[820] The_dragon, Modding Q&A.
[821] The_dragon, Modding Q&A.
[822] The_dragon, Modding Q&A.
[823] Waldzios, Modding Q&A.
[824] jacobhinds and kalarhan, Modding Q&A.
[825] Earendil redirect, Autocalc and Unit Levels. Get the normal links in the Forge Scrap Yard for better integration.

11.10 Default warnings of Native in rgl-log

Four default warnings in Native in rgl-log[826], could perhaps correct the warnings in a new MS version.

11.11 Banners

Perhaps some own subchapter at another place. setting banner and flag[827]

synchronized changes for banners[828]

banner offset[829]

11.12 Startup of Game

startup of game[830]

11.13 How can I skip the Character Creation for Quick Testing?

Work in the informations of the thread here[831]

11.14 How can I change the Battle Size?

Work in the informations of the threads of kalarhan[832] See also the issue note at the header operations GitLab for the explanations of Vetrogor.

11.15 Is the Game actually doing something with the Names of Scripts, Presentations, etc.?

Scripts names, presentations names and so on are only used for debugging and for mapping to hard-coded items, in example things starting with `game_` or `wse_`.[833]

11.16 The Usage and Usefulness of Decompilers

There was a big discussion about decompiler in the forum once and multiple little ones. Since the discussions about this topic are getting pretty fast pretty emotional - the topic was and still is coming

[826] Caba‘drin, Modding Q&A.
[827] dunde, Modding Q&A.
[828] Lav, Modding Q&A.
[829] Lav, Modding Q&A.
[830] Lav, Modding Q&A.
[831] Somebody and kalarhan, Skipping character creation for mod testing.
[832] Modding Q&A and Modding Q&A.
[833] cmpxchg8b, Modding Q&A.

up every few months - it was decided not to discuss it at all at the forum and later at the discord.[834]

There are two more known decompilers and they can be seen as helpful and essential tools until they are not anymore, meaning that they might be used with dubious intentions. As such a decision has been taken to not promote them. The ones who are in the community since a longer time and know how to script themselves will find them anyway and have to decide at their own if they want to use them or not. For experienced scripters they can be useful to study other modules, although at some point you know by yourself how to do most stuff and features are often very mod-setting specific. Newbies however should not be lead towards them because the cannot handle stuff anyway and it causes more mess than it helps them.

Decompilers do not generate py files as they have been before, you have no variable names inside the scripts except the one of the global variables. Additionally critical files like module_constants.py are missing, so you can't simply compile a decompiled file again. While you won't be able to compile it again you can however read the scripts and adapt them accordingly into your own mod. That mostly lures newbies deeper into getting lost if they are not used to work with the Module System.

It is nearly impossible to hide your own work if decompilers are available. That's the reason why some mods which are only server side don't publish their files at all since only the server needs it, otherwise they could give an advantage to players, especially at the competitive scene.

There are (to my knowledge) three obfuscators which are working against decompilers and which are the reason why the known decompilers don't work at them. Someone with enough passion and time could however still create a custom decompiler for any mod which used an obfuscator. The obfuscators are by Caesim, Swyter and The_dragon.

[834] One of the biggest discussions has been initiated by xenoargh, Bytecode MS Decompiler.

12 BRF Resource Management

12.1 What is OpenBRF and what are BRF Files?

BRF files are a file format used by the Mount&Blade game series to hold most game graphical content: meshes, materials, textures, collision boxes, shaders, animations and skeletons for animations. OpenBRF is a tool to edit and preview the content of those BRF files. You can preview them, import/export them in an array of formats (so that you can edit them in external applications), as well as edit them. This would be useful to build your own game module aka mod.[835] OpenBRF was originally developed for Mount&Blade version 1.011 (also called Vanilla M&B) but is backward compatible with Mount&Blade .808 BRF files (load only) and “forward“ compatible with Warband BRF files (save/load).[836]

For the basic usage: To access most functions you must select the object you want to act on (e.g. a mesh), clicking it on the list at the left, and then: either right click on it or use the [Selected] menu option, and then select the command from the menu. You can also select multiple objects, by Shift- / Ctrl-clicking on them. That way you can perform an action on all of them at once (via the context menu, as normal). Many things are also explained in the context menus.[837]

12.1.1 BRF File Format

If you save the brf file as Warband Resource you have to make sure everything has the ’standard’ 30000 (or 30001 as used by buildings, it seems) flags, but if you save as M&B Resource it can work with 0 flags. Many folk have had problems with the Warband Resource .brf type and many people have had problems of the game simply not starting if you have meshes with 0 flags.[838] If all meshes have 30000 or 30001 flags then the Warband resource works as well.[839]

rsf files[840]

12.1.2 Loading Order

Load order skeletons and animations[841]

[835] mtarini, FAQ and Troubleshooting.
[836] mtarini, Download link and main info.
[837] mtarini, FAQ and Troubleshooting.
[838] FrisianDude, Modding Q&A.
[839] FrisianDude, Modding Q&A.
[840] armagan, Mesh .rsf formats? What are the character maps for?.
[841] dstn, Mount & Blade Modding Discord.

12.2 Meshes

Limit of vertices per mesh[842]

hard limit vertices per mesh[843]

get size of shields[844]

Highelf old tutorial for triangle count reference[845]

12.2.1 Multi-meshing

When looking through the brf files of various mods you might have discovered the possibility to let a mesh consist out of multiple meshes. One of these might look like as in the screenshot below or the following:

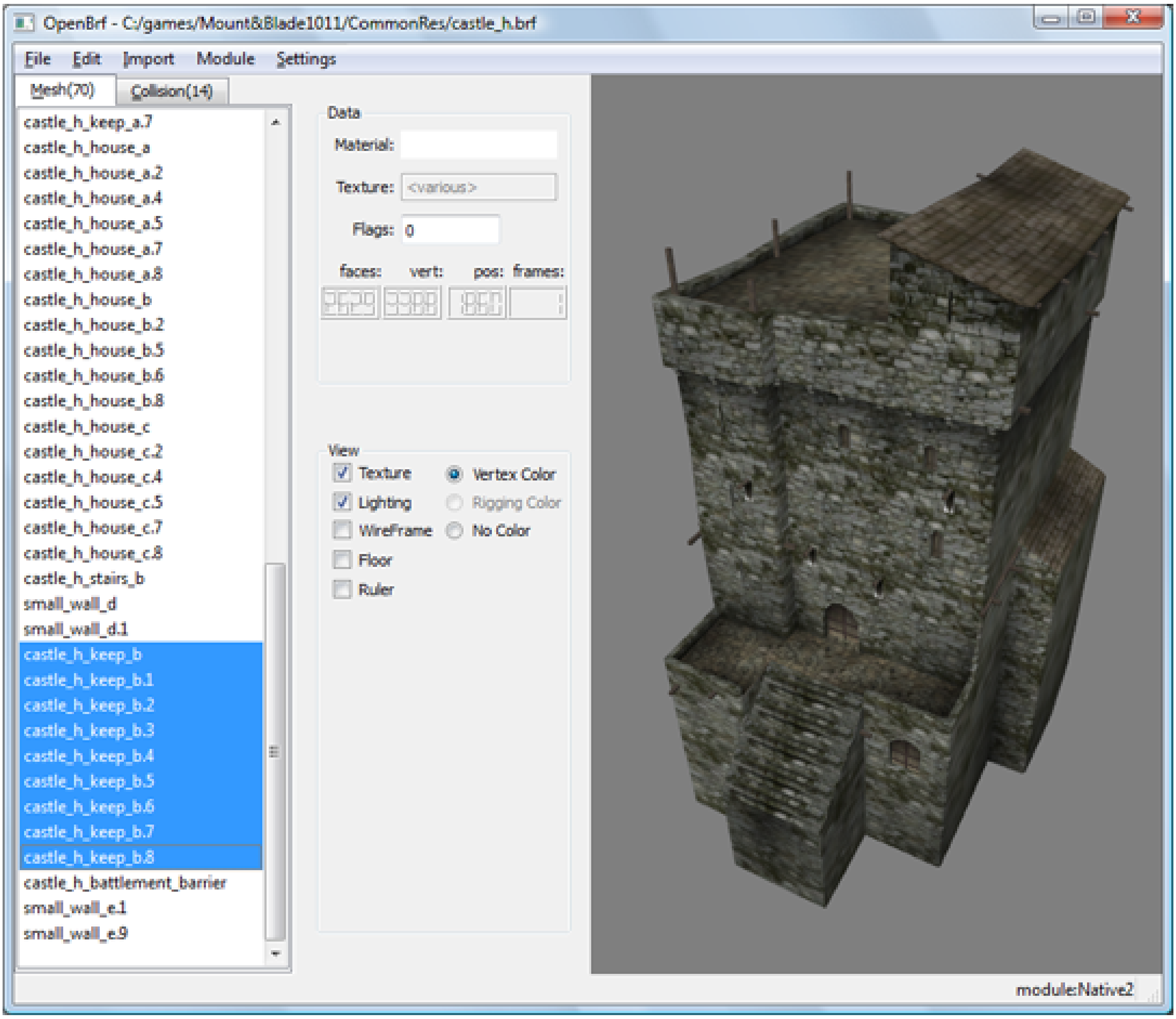
```
mesh_name
mesh_name.1
mesh_name.2
```

This is the so called **multi-meshing**. Multi-meshes work because the game engine thinks of a list of meshes with a .# suffix as a single unit and will combine those meshes on its own. You might be asking yourself why you shouldn't simply combine those meshes into one. That is because every mesh is limited to one material (and thus one texture set).[846] At some weapons or especially scene props it can however be sometimes necessary to use different materials on multiple parts of the same mesh, resulting in multi-meshing it.

With the latest version of Warband you can pretty much multi-mesh everything, from items to scene props.[847] At versions earlier than 1.134, so Vanilla M&B of course as well, multi-meshing is only possible for scene props and flora, not for items. Latter one will display only the first mesh from the list.[848] You will still face this old restriction in a similar way at another related field: multi-meshed items will be automatically combined when displayed as weapon icons, using the first mesh's material definition[849] and the inventory view only displays the primary mesh.[850] It is therefore still worth to think about combining item meshes by creating texture atlases, re-uvmapping the mesh so that all parts are using the same material.

[842] cmpxchg8b, Global Map Visualiser.
[843] cmpxchg8b, Modding Q&A.
[844] Somebody, Modding Q&A.
[845] Highelf, All about 3d-modeling, Chapter 1 : basic modeling.
[846] Swyter, Download link and main info!.
[847] Swyter, Download link and main info!.
[848] GetAssista, Modding Q&A.
[849] Somebody, Combining meshes/models in OpenBRF question.
[850] Somebody, Modding Q&A.

Another drawback is that every multi-mesh will need to load and apply multiple materials, textures and/or shaders which will affect the performance.[851] It is assumed that the engine has to render every texture even if they are hidden behind something. This gets worse the closer the player gets, meaning more pixels in the screen-space have to render the expensive shaders that come with armour for example. You might have experienced this when looking at large amounts of guys melee that the fps changes a lot depening on if you watch them in melee or are looking away of them. Reducing the number of multi-meshed items has been found to be good for performance, any more than three or four meshes on an item can start to kill fps due to draw calls.[852]



castle_h_keep_a.7
castle_h_house_a
castle_h_house_a.2
castle_h_house_a.4
castle_h_house_a.5
castle_h_house_a.7
castle_h_house_a.8
castle_h_house_b
castle_h_house_b.2
castle_h_house_b.5
castle_h_house_b.6
castle_h_house_b.8
castle_h_house_c
castle_h_house_c.2
castle_h_house_c.4
castle_h_house_c.5
castle_h_house_c.7
castle_h_house_c.8
castle_h_stairs_b
small_wall_d
small_wall_d.1
castle_h_keep_b
castle_h_keep_b.1
castle_h_keep_b.2
castle_h_keep_b.3
castle_h_keep_b.4
castle_h_keep_b.5
castle_h_keep_b.6
castle_h_keep_b.7
castle_h_keep_b.8
castle_h_battlement_barrier
small_wall_e.1
small_wall_e.9

Material:
Texture: <various>
Flags: 0
faces: 2629 vert: 3388 pos: 1850 frames: 1

☑ Texture ● Vertex Color
☑ Lighting ○ Rigging Color
☐ WireFrame ○ No Color
☐ Floor
☐ Ruler

Take note that multi-meshing is not possible for map icons. You need to work around this by using two map icons.[853] However, multi-meshing is possible for LODs, given you apply a name convention

[851] Somebody, Modding Q&A.
[852] jacobhinds, FormationsAI Performance Improvements/Stutter reduction?.
[853] Specialist and Somebody, Modding Q&A.

as follows:[854]

```
mesh_name
mesh_name.1
mesh_name.2
mesh_name.lod1
mesh_name.lod1.1
mesh_name.lod1.2
...
```

12.2.2 LODs, the Level of Details

Lods and optimization[855]

lods for buildings[856]

lods for submeshes[857]

lods distance triggers (move up to game engine chapter perhaps)[858]

Lods optimal amount[859]

What are lods[860]

set range of lod effect[861]

12.2.3 Rigging

Gloves are not rigged but vertex-animated. They don't have enough bones to relate to to make rigging useful.[862]

gloves and rigging[863]

old gauntlet frames[864]

gauntlet stuff[865]

gloves have sometimes this special stuff with not only _R (needed for unrigged gloves) but also expansions with _Rx[866]

[854] Docm30 (credit list?), Modding Q&A.
[855] GetAssista, Modding Q&A.
[856] MadVader, Modding Q&A.
[857] Gambino (credits), Modding Q&A.
[858] cmpxchg8b, Modding Q&A.
[859] jacobhinds, Modding Q&A.
[860] troycall, Modding Q&A.
[861] ealabor, How to set the range at which lods trigger.
[862] GetAssista, Modding Q&A.
[863] Mammoet, Modding Q&A.
[864] Somebody, Modding Q&A.
[865] rgcotl (credit), Modding Q&A.
[866] dstn, Mount & Blade Modding Discord.

rigging head/neck[867]

gloves rx lx[868]

12.2.4 Vertex Animations[869]

A vertex animation is just a sequence of meshes, one mesh per frame. Each frame is also associated to a timing. But here is the catch, each frame only redefines positions and normals.[870] All frames share the same: triangles (which triangle connects which vertices), texture coords (coordinates),[871] rigging (in case the mesh is also rigged), and so on. This also means that the frames are bound to have the same number of vertices and faces.

Note that the frames must be considered just keyframes. The game is "free" to interpolate between two frames of the vertex animation to get smoother animations. It will exploit that, for example when it animates bows.

Vertex animations are used for many things:

- Animated map icons on the world map, in which case the first frame[872] is unused, the second is the stance pose, the rest is the walking sequence.
- Weapons animations, like the bow and crossbow strings.
- Armours, at which the first frame is unused, the second frame is the male and the third the female version. Take note of the time of frame 2, the female one. The necessary time here gets determined by the morph key of the skin entry. Look up chapter 5.3.2.2 for more informations about skin flags and morph keys.
- Quivers and the like, at which each frame represents a different number of arrows, bolts, etc. which are still left.
- Shields, at which a frame is for when they are fastened around the body and the other one for when the belt is loose and the shield gets carried at the arm.
- Scabbards, at which the first frame is with sheathed weapon and the second frame without it.
- Hands or gloves, at which the first frame is an open glove, the second half-closed and the third closed in a fist or grab.

[867] Ivkolya (credit), Modding Q&A, Modding Q&A and Modding Q&A.

[868] Somebody, [WB] Warband Script Enhancer v3.2.0.

[869] Most informations taken from mtarini, Download link and main info!.

[870] Normals as in the perpendicular vector of each vertex, used to calculate lighting. When vertex animation happens, vertices have to move and thus recalculate their normals; jacobhinds, Mount & Blade Modding Discord.

[871] Meaning that it's not possible to have different frames with different UVs; jacobhinds, Mount & Blade Modding Discord.

[872] The numbering at frames starts with 0, so the first frame is frame 0.

- Customizable heads, at which frame 0 is the base head. Each frame is the effect of a “slider“, like “eye-to-eye distance“.
- Scene props, see chapter 5.4.1.5 for more details.

... and maybe other things too.

You have two options for making your own vertex animations in OpenBrf. Since version 0.38 of the tool, you can import and export vertex animations as MD3 files, the format introduced by Quake3. Alternatively you stack them frame by frame. So, typically, you export a frame, edit it, reimport it and attach it to your animation.

You can do this in the cut-and-paste way: with ctrl-alt-x, ctrl-alt-c, ctrl-alt-v (as you can see in the edit menu) you perform cut and paste operations that refer to individual frames (and apply to the currently selected frame). So you can import a mesh and stack it to your animaton by cut and paste. You can also redefine the frame order, duplicate frames, etc.

Otherwise you can do it via direct frame import and attach: when your frame is in an external file, as a shortcut, you can also import a static mesh directly as an additional frame of the currently selected mesh (menu ïmport").

At all methods, remember to edit at the end of the process manually the times (in the "data"box) of each frame.

What makes the process messy (if you tried in other occasions, like in OpenBRF you know) is that when you assemble frames you are assuming that not only the vertex number, but even their internal order has been kept the same by all import and export operations. Otherwise, say vertex N.412 is on the head in frame 1, but on the foot in frame 2: then it kills everything! Textures, triangles, interpolations... result is a mess. That is because, as you might remember, triangles, texture coords, etc. are shared by all frames. So, if you have a triangle connecting vertex N.412, N.51 and N413 together, that applies to **all** frames.

In OpenBRF you have two options to merge a frames into an animation. You select which one of them below is your favourite option, 1 or 2, in the option menu (“Option“ => “On assemble vertex animation“).

Option 1: you pray that the vertex order is kept unmodified by your mesh import/exports (OpenBrf will comply, but who knows what, say, Blender or Wings3D will do). This is the “old way“ and it works, sometimes.

Option 2: you resort to use texture coords as unique identifiers of vertices. You just assign to each vertex of your mesh an unique UV-point in texture space.[873] Once your texture coords identify

[873] Note that sometimes you want different points to share the same texture coords: for example, if you have a quiver,

the vertices in frame 1, don't touch them anymore. Feel free to edit the mesh 3D shape for all other frames!

Note that the system is somewhat robust: if option 1 fails (e.g. different number of vertices) it will fallback to option 2 automatically. If option 2 fails (e.g. same texture coords for different points), it will fallback to option 1 (for that vertex only).

Work in remaining notes if not already mentioned anyway:

vertex animate items[874]

vertex animating sword scabbard[875]

vertex animation[876]

sword scabbards[877]

merge mesh vertex animation[878]

moving rotating scabbard mesh[879]

firearms no vertex animations[880]

sword scabbards[881]

12.2.5 Vertex Colouring

Color uniform option at OpenBRF.[882]

get vertex colours into openbrf[883]

12.2.6 Others

hard-coded default bullet flying mesh.[884]

Reskeletonize meshes. Add reference to SSkin Flag and Morph Key"chapter.

each arrow is a copy of each other arrow, including the UV-coords, and if you have a symmetric something, the left half will use the same texture coords as the right half. In these cases, just displace the texture of each vertex by a little tiny bit in your mesh editor. Half a fraction of a texel (a texture pixel); it won't make any practical difference, but it will be enough for OpenBRF to tell vertices apart.

[874] Lumos, Modding Q&A.

[875] SupaNinjaMan, Modding Q&A.

[876] mtarinie (find original post), Modding Q&A.

[877] quote, should already be present above somewhere, Modding Q&A.

[878] without real solution but I have had that case, dig out at Modding Discord, Modding Q&A.

[879] Ivkolya, Modding Q&A.

[880] Mike12096, Modding Q&A.

[881] Swyter, Download link and main info!.

[882] Somebody, Modding Q&A.

[883] Vornne, Modding Q&A.

[884] ithilienranger and SonKidd, Modding Q&A.

12.3 Materials

prevent sun from going through flora[885]

don't block light at materials? Sun shining through sceneprop (interesting side bug with golden shining river)[886]

semi-transparent meshes (brf bug?)[887]

render order[888]

12.4 Textures

Average texture limit[889]

Specular map[890]

square texture size[891]

Mipmaps[892]

.lod textures[893]

alpha performance[894]

low-res option uses mip-maps[895]

alpha stuff at texture[896]

Older version gave error message at wrong size of texture[897]

Interesting, texture size etc[898]

textures not found (white-blue checkerboard patter)[899]

The colour `8080FF` is the base colour for normal maps and makes them flat.[900]

12.5 Collision Meshes

Multi-meshing is not possible for collision-meshes.[901]

[885] _Sebastian_, Modding Q&A.

[886] Ruthven and jacobhinds, Modding Q&A.

[887] _Sebastian_, Modding Q&A.

[888] Lumos, Modding Q&A.

[889] xenoargh, Modding Q&A.

[890] jacobhinds, Modding Q&A.

[891] jacobhinds, Modding Q&A.

[892] _Sebastian_, Modding Q&A.

[893] jacobhinds, Modding Q&A.

[894] Somebody, Modding Q&A.

[895] FantasyWarrior (credit), Modding Q&A.

[896] SupaNinjaMan, Modding Q&A.

[897] Chilly5, BaldwinIV and fisheye (credit), dds error: size=not a power of 2.

[898] mtarini, Shader and Graphical effects Discussion.

[899] mtarini, FAQ: textures not found (white-blue checkerboard patter): how to fix?.

[900] jacobhinds, Mount & Blade Modding Discord.

[901] GetAssista, Modding Q&A.

BRF Format and collision meshes[902]

12.6 Skeletons

hitboxes for skeletons[903]

12.7 Animations

T frame at new animation[904]

12.8 Performance Optimization

Perhaps integration at other subchapters, will see how it goes.

Optimize performance of a mod[905]

performance question here is a loading time question[906]

Mod optimization [907]

Loading stuff[908]

Loading overhead 10 items vs one new texture[909]

rearranging brf files for optimization[910]

resource management[911]

brf organisation[912]

philosophy and optimisation[913]

optimisation[914]

common errors at brf resource management[915]

Texture atlas of 4k recommended and some more optimisation stuff[916]

[902] fisheye, BRF Format and collision meshes: brief documentation (Pt 1).

[903] La Grandmaster, Modding Q&A.

[904] NPC99, Modding Q&A.

[905] Abhuva, Modding Q&A.

[906] Somebody and cmpxchg8b, Modding Q&A.

[907] Caba‘drin, Modding Q&A.

[908] Caba‘drin, Modding Q&A.

[909] Somebody, Modding Q&A.

[910] kraggrim, Modding Q&A.

[911] kraggrim, Modding Q&A.

[912] _Sebastian_, Modding Q&A.

[913] kalarhan, Modding Q&A.

[914] _Sebastian_, Modding Q&A.

[915] Ramaraunt (credit), Modding Q&A.

[916] Marko, Leonion and NPC99, Modding Q&A, Modding Q&A and Modding Q&A.

12.9 Remaining notes

Some map icon/heraldic informative stuff[917]

some useful brf info (normal map -> tangent blubb), also follow up posts[918]

realistic shadow on plants[919]

morph not working for boots[920]

import static meshes[921]

openbrf copy complete (mesh+material+textures)[922]

Don't forget Maroon's Tutorial and OpenBRF reading marks.

tangent directions[923]

12.10 Shaders

Shaders[924]

Some shaders don't support vertex animation for quivers[925]

Adding shaders, perhaps outdated[926]

Some shader notes[927]

Interesting stuff with animated textures, perhaps dstn will test it[928]

water reflections[929]

Skinned shaders[930]

billboard shaders (need to ask dstn about optimality)[931]

shaders stuff[932]

engine picks random texture for shader if not provided[933]

green and pink spots at terrain (some problem dstn tried to figure out, shader related?)[934]

[917] dunde, Modding Q&A.
[918] NPC99, Modding Q&A and Modding Q&A.
[919] Ruthven, Modding Q&A.
[920] cmpxchg8b, Modding Q&A.
[921] Swyter, Modding Q&A.
[922] Swyter, Modding Q&A.
[923] mtarini, Recompute Tangent Dirs.
[924] mtarini, Shader and Graphical effects Discussion.
[925] GetAssista, Modding Q&A.
[926] Abhuva and Vornne, Modding Q&A.
[927] SonKidd, Modding Q&A, and mtarini, Green Normalmaps.. Question??.
[928] Thorgrim, New version texture errors.
[929] Vornne, Modding Q&A.
[930] jacobhinds, Modding Q&A.
[931] jacobhinds and MadocComadrin, Modding Q&A and Modding Q&A.
[932] jacobhinds, Modding Q&A.
[933] _Sebastian_, Modding Q&A.
[934] Swyter, old forum link, Shader Stuff- HLSL instruction limits.

beard problem with shader (not happening with Native ones)[935]

Seems like using non-Native shaders inside of a skin's .LOD material causes crashes when the LOD material is used.[936]

shader stuff[937]

Tutorials

Page 4 of Jiks documentation with personalizing your mod. Maybe making an easy walk through with jiks example? Starting with the hard-coded menu tutorial for first success. Afterwards easy first steps with adding new stuff at each py file. Reread jiks integrated tutorials for that to integrate them properly.

Before we begin, we will take the first step in personalizing your mod. Let's change the mod selection picture. This can be done with MS Paint, or anything that will edit BMP files. The file **main.bmp** is located in your Mount&Blade modules folder - for example in C:\Program Files\Mount&Blade\Modules\MyMod. Editing this will change the picture shown when you select your mod. This is often a good idea if you jump between working on a mod and playing others.

If you have the ability to edit DDS image files, you can make copies of the various DDS background pictures to your mod's texture folder - for example from C:\Program Files\Mount&Blade\Textures to C:\Program Files\Mount&Blade\Modules\MyMod\Textures. If you change them in your mod's texture folder, they will be changed in the game when you play your mod. The file **bg2.dds** is the standard background for most menu pictures, and **pic_mercenary.dds** is the picture for the main menu. Don't spend too much time with this now, but know that you can make your mod look a lot slicker when it's done.

game hints[938]

Remaining notes for sorting

Create variables that point to registers.[939]

language files messages (Lav extra py file)[940] ui cvs stuff (Lave WRECKER has module file

[935] Garedyr (credit), Modding Q&A.

[936] dstn, Mount & Blade Modding Discord.

[937] Swyter, [WB] Warband Script Enhancer v3.2.0.

[938] Caba'drin, Modding Q&A, and Somebody, Modding Q&A.

[939] MadocComadrin, Modding Q&A.

[940] Seek n Destroy, Modding Q&A.

for them)[941] change loading text[942] hard-coded ui text[943] change names of attributes[944] language folders[945]

Disable Steam achievements[946]

debug mode at game and other useful informations afterwards[947]

Empty folder to clear Warband memory, need to find more info here[948]

compiling at Linux[949]

A part of history, gk_parry_then_attack[950]

Slot agent is blocked[951]

Change Combat Feed Font Size[952]

Ghost range of shields[953]

Swyter doing some calculations[954]

ai targeting hard-coded? and turn of archer ai (next page too)[955]

ignore specific key to be pressed and zoom (requires module.ini setting)[956]

optimization at code[957]

Legendary cow bug[958]

Disallow horses (and no cavalry bots) (tf mounted mentioned for mp)[959]

load game faster with fps setting[960]

Target a group for priority fire (game engine)[961]

stationary troops at moving ships problem[962]

horse archer ai (MS Overhaul?)[963]

[941] Somebody, Modding Q&A.
[942] Seek n Destroy, Modding Q&A.
[943] kalarhan, Modding Q&A.
[944] Leonion, Modding Q&A.
[945] kalarhan, Modding Q&A and Modding Q&A.
[946] Somebody, Modding Q&A.
[947] Swyter, Modding Q&A.
[948] Chatter, Modding Q&A.
[949] kalarhan, Modding Q&A.
[950] Ruthven, Modding Q&A.
[951] kalarhan, Modding Q&A.
[952] DeathByRabbit (credit), How to change the Combat feed/Font size.
[953] SupaNinjaMan, Shield Hitbox Issue.
[954] Swyter, Modding Q&A.
[955] Multiple people at discussion, Modding Q&A.
[956] _Sebastian_, Modding Q&A.
[957] gsanders 8credit, Modding Q&A, and Leonion, Modding Q&A
[958] Dalion, Modding Q&A.
[959] Ivkolya, Modding Q&A, Modding Q&A and Modding Q&A.
[960] Pitch, Modding Q&A.
[961] kalarhan, Modding Q&A.
[962] dommipoppe, Modding Q&A.
[963] Diverse, Modding Q&A and Modding Q&A.

very basic permadeath[964]

key commands for main menu[965]

interesting attach sceneprop fail[966]

agent getting ridden by another agent[967]

Random number engine generates even numbers?[968]

crouch crossbow reloading, open question[969]

Storing and retrieving binary flags as integers[970]

Optimizations[971]

integer size, not sure how to implement the best[972]

warband multicore[973]

WSE a bit explained[974]

How to add new operation (game engine)[975]

128 pos (regs etc too?)[976]

interesting info bit[977]

item drop at death[978]

[964] Ruthven and kalarhan, Modding Q&A.

[965] Winter, Pleasant surprise for modders.

[966] Dawiduh, Modding Q&A and Modding Q&A.

[967] Lumos, Modding Q&A, and Veni Vidi Vici, Mount & Blade Modding Discord.

[968] Veledentella, python code doesn't work.

[969] Derek_Custer, Modding Q&A.

[970] fujiwara, Storing and retrieving binary flags as integers.

[971] Multiple participants covering multiple areas, Optimizations.

[972] cmpxchg8b, Modding Q&A, and Vornne, Modding Q&A.

[973] K700, Warband Script Enhancer 2 (v1.0.4.6).

[974] Ra'Jiska (credit), [WB] Warband Script Enhancer v4.8.0 for 1.174.

[975] kalarhan, Modding Q&A.

[976] cmpxchg8b, [WB] Warband Script Enhancer v3.2.0.

[977] cmpxchg8b, [WB] Warband Script Enhancer v3.2.0.

[978] cmpxchg8b, [WB] Warband Script Enhancer v3.2.0.