

第二章：MOD 內容

MOD 項目

現在，已經把基本配置弄完成了，點開 Module_system 1.171 文件夾。在文件夾裡，有一個 build_module 的 bat 批次檔（也是 PY 程式編寫的）和 PY 檔案。在 PY 檔案裡，有分四種文件：

- header_前綴文件
- process_前綴文件
- ID_前綴文件
- module_前綴文件

header_文件，是屬於 MOD 中的常數文件，可以去修改的文件，但是在未深入開發時則不建議去變動，所有操作的函數歸類在 header_operations.py 文件內。如果真的要定義新的向量，可以在 module_文件修改。

process_文件，主要是 MOD 編譯成數據（數字），因為 process_文件是屬於重要的文件，在未確認他的運作流程時千萬別去更動。

ID_文件，主要是運作遊戲內部的單元，是遊戲內部所有事項的索引區塊。是生成 MOD 時所建立的臨時文件，在每次的編譯時會重新創建，所以不能去修改，也不能在編譯前將 ID_文件刪除。你也能夠利用 ID_文件去檢查所有事項的索引，如果遊戲出現 BUG 錯誤，會出現錯誤 BUG 的條列索引，可以在 ID_文件中快速查獲。

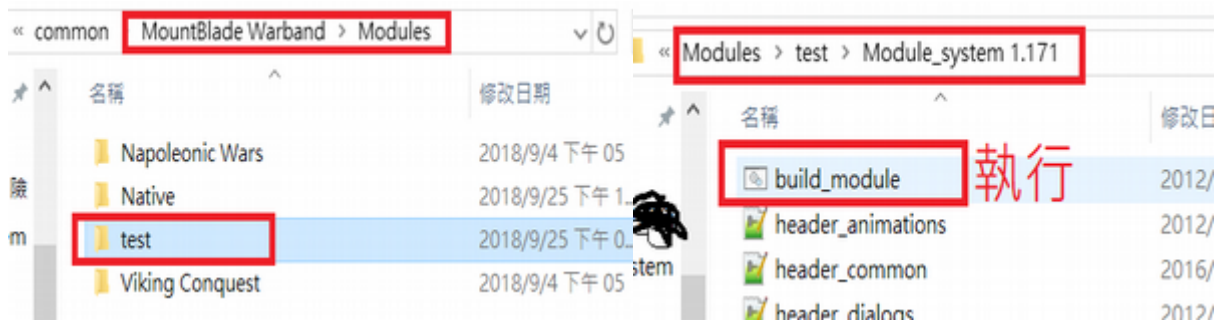
module_文件，包含的是 MOD 內容，是我們主要修改的文件。包含修改兵種、物品、裝備、NPC、對話、場景等等。

＊（除了 process_文件和 build_module 的 bat 批次檔是 python 語言去編寫，其他的文件都是用 MS 語法去編寫的。）

最後就是 build_module 的 bat 批次檔，他是最主要的編譯執行檔案。執行這類的檔案，它會自行將 py 檔案編譯成 txt 文檔，並覆蓋原先 MOD 的 txt 文檔。

MOD 編譯

現在，把原版 MOD（Native）文件夾複製，然後貼到 Modules 文件夾中，改檔名為 test。
再來是進入到 test\Module_system 1.171 文件夾裡，執行 build_module 的 bat 批次檔。



在未修改裡面任何的文件下去執行，結果出現了錯誤。先查看 module_info.py 文字內部代碼，有一行是 `export_dir = "../WOTS/Modules/Native/"`，以我的 MOD 為例，放置 MOD 的位置在 `D:/Program Files (x86)/Steam/steamapps/common/MountBlade Warband/Modules/test/`，那就要修改 `export_dir` 指向文件夾，修改成 `export_dir = "D:/Program Files (x86)/Steam/steamapps/common/MountBlade Warband/Modules/test/"`。

*若是檔案路徑過長，或者是嫌太麻煩，可以修改成 `export_dir = "../"`，指向預設的目錄。

*在 MOD 系統裡，目錄名是用正斜槓(/)而不是用反斜槓(\)，再來就是目錄尾端還要在加上一個正斜槓。

```
C:\Windows\system32\cmd.exe
initializing...
variables.txt not found. Creating new variables.txt file
Compiling all global variables...
variables.txt not found. Creating new variables.txt file
variable_uses.txt not found. Creating new variable_uses.txt file
Traceback (most recent call last):
  File "process_global_variables.py", line 106, in <module>
    save_variables(export_dir, variables, variable_uses)
  File "D:\Program Files (x86)\Steam\steamapps\common\MountBlade Warband\Modules\test\Module_system 1.171\process_operations.py", line 171, in save_variable
    file = open(export_dir + "variables.txt", "w")
IOError: [Errno 2] No such file or directory: '../WOTS/Modules/Native/variables.txt'
Exporting strings...
Traceback (most recent call last):
  File "process_strings.py", line 26, in <module>
    save_strings(strings)
  File "process_strings.py", line 9, in save_strings
    ofile = open(export_dir + "strings.txt", "w")
IOError: [Errno 2] No such file or directory: '../WOTS/Modules/Native/strings.txt'
Exporting skills...
Traceback (most recent call last):
  File "process_skills.py", line 32, in <module>
    save_skills()
  File "process_skills.py", line 15, in save_skills
    ofile = open(export_dir + "skills.txt", "w")
IOError: [Errno 2] No such file or directory: '../WOTS/Modules/Native/skills.txt'
```

之後再執行一次 build_module 的 bat 批次檔，如果運行解果如下面的輸出文字所示，那就代表執行成功。

```
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...
Exporting mission_template data...
Exporting game menus data...
exporting simple triggers...
```

exporting triggers...
exporting dialogs...
Checking global variable usages...
Exporting postfx_params...

Script processing has ended.
Press any key to exit. . .

*一般按照上面的流程做下去，是不會出現錯誤的。但如果其他的錯誤訊息，在官方和中文站上有著各種 BUG 的解決方法，請善用搜索引擎。

[分享]收集：常見 py 報錯解決方案（4-17 更新）：by 惡豬

<http://bbs.mountblade.com.cn/thread-487284-1-1.html>

[分享]關於製作 MOD 時常見的 bug&以及其解決方法 by 心中飛翔

<http://bbs.mountblade.com.cn/thread-360772-1-1.html>

編譯完成了，可以到 MOD 的 txt 文檔看看，除了 `game_variables.txt` 和 `map.txt` 沒更新外，其他的 txt 文檔都已經更新。

MOD 文檔項目細則

在 MOD 的編譯文檔中，雖然有分著四大類，但這四大類之中，每一類又分著不同的項目。在這四大類的各個項目，雖然有許多是相同的項目，但是也有的是單一種類，下面的條目是用來分類的細則。

module 文檔

項目	說明
animations	動作編輯（順序和總量不能更改）
constants	變量，slots 和其他變量都能在這設定
dialogs	對話（單機）
factions	勢力
game_menus	遊戲介面菜單
info	指向輸出目錄（輸出 txt 檔路徑）
info_pages	記錄裡遊戲概念的文字內容
items	物品和裝備的各項設定
map_icons	大地圖圖標
meshes	圖片定義。例如進入遊戲時的背景圖片，及盾牌上的紋章圖案
mission_templates	戰鬥場面的定時觸發器（聯機主要修改內容）
music	音樂
particle_systems	粒子系統。如火焰、閃電、煙霧等等
parties	大地圖單位，包含建築、部隊、領主等單位的定義
party_templates	大地圖上的部隊模板定義
postfx	著色器參數設定
presentations	定義介面顯示效果
quests	任務，只定義名稱（單機）
scenes	場景定義，如地形、出生點、水平高度等
scene_props	場景物品，如攻城器具（要有模型）
scripts	功能（程式）模塊集中地，讓各大區塊能夠調用

simple_triggers	大地圖定時觸發器（舊版簡單觸發）
skills	技能定義
skins	皮膚、臉及身體定義
sound	音效
strings	字串語句，如進遊戲開新局時出現的介紹字串
tableau_materials	模型材質參數，還有紋章盾牌、衣服等
triggers	大地圖定時觸發器（新版複雜完整觸發）
troops	兵種（升級樹）、領主、國王和 NPC 等定義
variables	變量（系統）

header、process 和 ID 文件

項目	說明
header_common	公共常數（聯機）
header_ground_types	場景類型常數
header_item_modifiers	物品修飾名常數
header_operations	操作常數
header_terrain_types	地勢類型常數
process_global_variables	全局變量處理
process_global_variables_unused	無用全局變量處理
process_init	初始處理
process_line_correction	行列矯正處理
process_particle_sys	粒子系統處理
process_party_tmps	部隊模板處理
process_tags_unused	無用標籤處理
ID_postfx_params	著色器變量 ID

MS 源碼大致規格

打開任一個 module_ 的文檔，除了有少數檔案，大致有著三個規格。



引進模組	騎馬與砍殺的 python 腳本，彼此都有引入或者是被引入到各自區塊的程式碼，作為引用參數的模組。（可以了解代碼間的相互關係性）
代碼講解	第二個區塊是用井字號包圍的綠色英文，是講解下面的源代碼大致情況，以及如何簡略的修改參數。（代碼講解很重要，要學習 MS 代碼的一定要看）
源代碼	就是我們要修改的項目。

整理 MS 檔案歸檔

這個部分我是參考 rubik 的領軍者這部分，關於領軍者的 MOD 夾帶著原代碼，下載地址：

【領軍者：技能大修】for 戰團 1.171 版及以上，2018-3-10 更新 1.439 版 by rubik

<http://bbs.mountblade.com.cn/thread-120447-1-1.html>

將 Modules_system 1.171 檔案夾改成 source，在 source 檔案內新建一個檔案，名字改為 MS File，然後將 header_前綴文件、ID_前綴文件和 process_前綴文件全部丟進 MS Files 檔案內。接下來就是要修改文檔，而文檔主要修改的部分就是 **build_module.bat 批次檔**。

點選 build_module.bat 批次檔，然後以記事本或者是 Notepad++ 文檔編輯器方式開啟，在 **@echo off** 的下一行中輸入：

```
@del Process_Log.txt
```

```
copy ".\MS Files\*" "*" ">>Process_Log.txt
```

再來是在 **python process_postfx.py** 下一行中輸入：

```
@del header_*.py
```

```
copy ".\ID_*.py" ".\MS Files\" ">>Process_Log.txt
```

```
@del ID_*.py
```

```
@del process_*.py
```

存檔後執行就完成了，這個檔案歸檔主要是為了方便查看檔案資料。

*一開始的執行可能會出現 Process_Log.txt 找不到的訊息，不過不影響輸出結果，再次執行就沒這問題了。

程式註解

在 MS 腳本中，許多文件都是上千上萬行的程式碼，一般外行人要理解需要花上相當漫長的時間。程式註解的主要原因是為了讓程式更容易了解，方便自己或者是其他人。

一般 python 的註解都是用#來註釋，但如果要大段落的註解，則是用三個單引號或者是雙引號包夾。註釋區在編譯時會忽略不去編譯。

```
'''
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 manage.
'''
單引號段落註釋

#ITEM SLOTS #號註釋

slot_item_is_checked = 0
slot_item_food_bonus = 1
slot_item_book_reading_progress = 2
slot_item_book_read = 3
slot_item_intelligence_requirement = 4
slot_item_amount_available = 7
slot_item_urban_demand = 11 #consumer demand for a good in town, measured in abstract
slot_item_rural_demand = 12 #consumer demand in villages, measured in abstract units
slot_item_desert_demand = 13 #consumer demand in villages, measured in abstract units
程式碼右邊註釋
```

在 PY 程式碼加上中文字幕

一般來說，python 在 3 版以前是不默認中文字幕的，裡面有中文字幕（就算是在註釋區輸入中文字幕）編譯時也絕對會出錯。

如果要在裡面輸入中文的話，就要在程式的首行（第一行）輸入`#coding=utf-8`，就可以在裡面輸入中文了。不過要編譯的遊戲內直接出現中文，這方法是不可行的。有關於更詳細的解決方法，在下方的帖子可以參考。

在 PY 代碼中直接加入中文對話 by alixyang

<http://bbs.mountblade.com.cn/thread-85836-1-8.html>