

# 插件

w3x2Ini支持自定义插件，你可以用插件来修改转换文件的行为。

!> w3x2Ini不会对插件行为做安全检查，进行一些错误的操作可能导致转换出错。

## 快速开始

在w3x2Ini根目录新建plugin文件夹，在里面放一个.config文件，写上想要加载的插件名，w3x2Ini便会去加载plugin\插件名.lua。

```
plugin\.config
```

修改单位名

```
plugin\修改单位名.lua
```

```
local mt = {}

mt.info = {
    name = '修改单位名',
    version = 1.0,
    author = '最萌小汐',
    description = '将所有单位的名字加上前缀"被插件修改过的"。',
}

function mt:on_full(w2l)
    for id, obj in pairs(w2l.slk.unit) do
        obj.name = '被插件修改过的' .. obj.name
    end
end

return mt
```

## 配置

配置文件为plugin.config，在该文件中写上想要加载的插件名，w3x2Ini便会去加载相应的插件。如果要加载多个插件，可以每行写一个插件名，w3x2Ini会按照顺序去加载这些插件。

```
插件1
插件2
插件3
```

## 插件

插件为`plugin\插件名.lua`，需要在`plugin\.config`中定义后才会加载。插件应该是一个lua脚本，它需要返回一张表，在表中可以有以下属性或方法：

## info

插件的基本信息，是一张拥有下列属性的表：

- `name` 插件的名字，字符串。
- `version` 插件的版本号，数字。
- `author` 插件的作者，字符串。
- `description` 插件的描述，字符串。

```
mt.info = {  
    name = '插件名',  
    version = 1.0,  
    author = '插件作者',  
    description = '插件描述',  
}
```

## on\_full

完整数据（Full）事件，关于完整数据的定义见[这里](#)。在该事件中可以简单方便的修改物编数据从而修改转换后的结果。

完整数据内的数据格式可以参考`data\zhCN-1.24.4\prebuilt\Custom`

```
-- 让所有技能无冷却无消耗  
function mt:on_full(w2l)  
    for id, skill in pairs(w2l.slk.ability) do  
        for i = 1, skill._max_level do  
            skill.cost[i] = 0  
            skill.cool[i] = 0  
        end  
    end  
end
```

## on\_mark

引用标记事件，在该事件中可以对对象的引用进行标记，以免转换Slk时对象被当做未使用对象而删除。这个事件期待返回一张表，这张表的所有`key`对应的对象都会被标记为引用。

```
-- 引用L000 - L009的对象，这些对象只在Lua脚本中使用，无法被自动引用  
function mt:on_mark()  
    local list = {}  
  
    for i = 0, 9 do
```

```

        list['L00'..i] = true
    end

    return list
end

```

## 接口

w3x2lni没有为插件准备专用的接口，而是将插件当做了代码的一部分，在调用插件的事件时将当前会话作为参数传入。也就是说，插件可以任意使用w3x2lni内部的函数，任意修改会话状态，你需要自己确保转换不会出错。这里提供一些常用的内部方法（假定传入的会话保存在变量w2l中）：

### slk

物编数据表，数据结构参考data\zhCN-1.24.4\prebuilt\Custom

```

for type, list in pairs(w2l.slk) do
    for id, obj in pairs(list) do
        for key, value in pairs(obj) do
            end
        end
    end
end

```

### setting

配置表，除了在config.ini中能看到的属性以外，还有以下属性：

- input (filesystem) - 输入路径。
- output (filesystem) - 输出路径。
- target\_storage (string) - 输出格式，mpq表示打包成地图，dir表示生成目录。
- mode (string) - 输出模式，slk、obj或lni。
- version (string) - 地图版本，Custom表示自定义地图，Melee表示对战地图。

```

if w2l.setting.mode == 'slk' then
end

```

### file\_save

保存文件

- type (string) - 文件类型，参考lni后的目录结构
- path (string) - 文件名
- buf (string) - 文件内容

```

w2l:file_save('scripts', 'blizzard.j', '')

```

## file\_load

读取文件

- type (string) - 文件类型
- path (string) - 文件名

```
local buf = w2l:file_load('script', 'blizzard.j')
```

## file\_remove

删除文件

- type (string) - 文件类型
- path (string) - 文件名

```
w2l:file_remove('script', 'blizzard.j')
```

## input\_mode

输入文件模式，**lni**表示输入地图是Lni模式的。

地图内插件

将**plugin**目录放在地图的**w3x2lni**目录中便是地图内插件，这里的地图既可以是目录格式（**Lni**）也可以是MPQ格式（**Obj**）。W3x2lni在转换该地图时便会应用地图内的插件。地图内插件会继续保留在转后的地图内，除非转换模式为**Slk**且启用了**删除只在WE中使用的文件**。