

本教程建立在 **ultrahand 版本 2.2.0 以上**。低于该版本，会有**很多函数不可使用**。建议使用最新版本。

另外注意，package 命令**不支持英文大写**！

ultrahand 扩展功能 packages（自定义插件包）的编写

packages 以下简称“插件包”，是一个**<英文命名文件夹>**

插件包文件组成

插件包在 SD 卡中的路径

`/switch/.packages/<英文命名文件夹>`

该目录的每一个**<英文命名文件夹>**都将被 ultrahand 视为一个独立的插件包。

package.ini

插件包的必要主程序，在 **ultrahand** 打开插件包时运行，位于：

`/switch/.packages/<英文命名文件夹>/package.ini`

boot_package.ini

非必要程序，位于插件包子目录，在插件包打开时会首先运行该 `boot_package.ini` 中[boot]这一节的命令，一般用来预加载一些内容，如从服务器获取最新版本信息。该文件中非"[boot]"节的内容不会自动运行，但可以在主程序中使用"exec"命令调取并运行。如果不存在[boot]节，则启动插件包时无事发生。

exit_package.ini

非必要程序，在 package 关闭/退出时，会自动运行该文件中[exit]这一节的命令。

config.ini

插件包中程序运行时自动生成的配置文件。

用于记录开关状态、页脚，也可灵活运用来记录一些需要频繁调取的动态参数等。

程序标题和特殊命令

每一节"[文本]"都将视为一个按钮/按钮分类标题。

[文字]

用于建立程序的标题或者按钮。单[]的按钮效果是，点击立刻运行他下面的命令行，无子页面。

[*文字]

表明该按钮为菜单形式，点击后根据下文定义的源文件/源数据来展开子页面菜单列表，通常由 source 相关的 ultrahand 命令定义列表内容。

[@文本 1]

[@文本 2]

插件包左右翻页按钮。

[@文本 1]和[@文本 2]之间的内容被视为第一页。

[@文本 2]之后的内容被视为第二页。

左右页切换通过十字键完成。

exec 文本

调取并执行 boot_package.ini 中的[文本]同名节的命令。

如果在后面加上文件路径，可以调用其它 ini 的命令。不用必须是 boot_package.ini。如

exec 文本 '.package.ini'

refresh

用于刷新当前的页面/页脚。

refresh package

重新加载当前 package.ini

refresh wallpaper

刷新/重载 ultrahand 的背景图，使其立即生效。

notify "文字" 21

notify-now "文字" 21

调用 ultrahand 的通知，在左上角显示文字内容。

加 **-now** 为即时弹出通知。可以用来覆盖上一个普通通知。

"21"表示文字显示大小，最大可设置 30。

set-footer 文本

为当前节的按钮设置页脚值，一般用于展开菜单中。

filter 文本

过滤命令，用于拉取多选项列表时，filter 后加空格写入你不想让他显示的选项。

/

指定文件路径所用的分隔符

单/为从根目录开始。

./为当前 ini 文件所在的目录开始，**../**为当前 ini 文件的父文件夹路径。

' "

单/双引号，当**完整的路径中有空格**时，为了防止命令行把他当成两段，需要用前后两个'让程序将其视为一个整体。

try

作用：尝试命令，该命令下的任何一条命令要是返回失败结果，那么整段命令会马上停止，运行下一个 try 命令。

path_exists

作用：检查指定路径是否存在，根据路径存在与否返回命令执行成功或失败状态，可在 try 块中链式调用实现条件判断。

用法：

path_exists <待检查的路径>

（如 path_exists /bootloader/hekate_ipl.ini 表示检查该路径下的 hekate_ipl.ini 文件是否存在）。

;title='文字'

设置包的标题名称，即最上方显示的名字

;creator='文字'

显示效果：

作者 Hahappify 忘忧

;color=#00AA99

标题的颜色设置

;version=

版本号信息，但后面随便你填啥。

这些 package 信息需要在页面最后使用信息显示命令调取才能显示。

;credits=

贡献者标题

系统命令

1. reboot

作用：重启系统。

也可用于重启进入 Hekate / UMS 模式，或通过启动项名称进入 Hekate ini 启动项。

用法：

reboot（普通重启）

reboot hekate（重启进入 Hekate 模式）

reboot ums（重启进入 UMS 模式）

reboot boot <启动项名称>（进入 hekate_ipl.ini 中的指定启动项）

reboot ini <INI 启动项名称> 或 reboot ini <INI 启动项名称> <重复索引>

reboot <Payload 二进制文件路径>（直接启动指定 Payload 文件）。

2. shutdown

作用：关闭系统或关闭所有蓝牙控制器。

用法：

shutdown（关闭系统）

shutdown controllers（关闭所有蓝牙控制器）。

3. backlight

作用：修改当前屏幕背光亮度值。

用法：backlight on（开启背光）

backlight off（关闭背光）

backlight <百分比整数>（设置背光为指定百分比，如 backlight 50 表示 50%亮度）。

文件系统命令

1. make / mkdir

作用：创建一个目录（文件夹）。

用法：

mkdir <目录路径>

```
mkdir '/config/ultrahand/new_dir'
```

表示在指定路径创建“new_dir”目录。

2. copy / cp

作用：复制单个文件或整个目录。

用法：

```
copy <源文件/目录路径> <目标文件/目录路径>
```

```
copy /sdcard/file.txt /config/ultrahand/
```

表示将“file.txt”复制到目标路径

3. mirror_copy / mirror_cp

作用：镜像复制目录内容（即完整复制源目录的所有内容，包括子目录和文件，保持目录结构一致）。

用法：

```
mirror_copy 空格<源目录路径>空格 <目标目录路径>。
```

```
mirror_copy '/atmosphere/contents' '/switch'（把 contents 文件夹和他里面的所有文件全复制到 switch 文件夹里。）
```

4. delete / del

作用：删除单个文件或整个目录。

用法：del <文件/目录路径>（如 delete /sdcard/old_file.txt 表示删除“old_file.txt”文件）。

5. mirror_delete / mirror_del

作用：以源目录的内容为参照，删除目标路径中与源目录内容匹配的文件（即仅删除目标路径中存在于源目录的文件）。

用法：mirror_delete <源目录路径> <目标目录路径>。

```
mirror_delete '/switch/dbi' '/atmosphere'
```

即为参照 DBI 文件夹内所有文件，删除 atmosphere 文件夹下所有同名文件

6. move / mv

作用：将文件或目录移动到新位置，或为文件/目录重命名。

用法：

`move <文件/目录路径> <目标目录路径>`

`move '/sdcard/old_dir' '/config/ultrahand/new_dir'`

表示将“old_dir”移动到目标路径并命名为“new_dir”

7. download

作用：从指定 URL 下载文件到本地目标路径，如果在目标路径的最后/文本，可以让下载后的文件直接改名。

用法：

`download <文件 URL> <本地目标文件路径>`

`download 'https://example.com/file.zip' '/sdcard/downloads/'`

表示将指定 URL 的压缩包下载到本地“downloads”目录

8. unzip（目标路径/后，加文件名称，可使文件改名）

作用：从指定 ZIP 压缩包中提取文件到本地目标路径，保持原始文件结构。

用法：

`unzip 空格 <ZIP 压缩包路径>空格 <本地目标路径>`

`unzip '/config/ultra/downloads.zip' '/'`

就是将 config/ultra 文件夹下的 zip 文件，解压到根目录

9. -filter（txt 文本的路径）

作用：从指定的 txt 文件里，获取需要过滤的文件与路径，使移动，复制，删除文件命令不对其生效。

示例：

`-filter ./123.txt`

txt 内容是：

atmosphere

这样，

`del '/atmosphere/'`命令就无法对大气层文件夹生效。

INI 文件命令

1. set-ini-val / set-ini-value

作用：编辑或创建 INI 文件，修改或者创建指定的区段和键值对。

用法：

set-ini-val <待编辑 INI 文件路径> <目标区段> <目标键> <目标值>

set-ini-val '/config/ultrahand/config.ini' '设置' '语言' 'zh-CN'

表示在“设置”区段下设置“语言”键的值为“zh-CN”

[设置]

语言=zh-CN

2. set-ini-key

作用：编辑 INI 文件，更新指定区段下的某个键名为新键名（键值保持不变）。

用法：

set-ini-key <待编辑 INI 文件路径> <目标区段> <原键名> <新键名>

set-ini-key '/config/ultrahand/config.ini' '设置' '文本' 'language'

把[设置]下的文本=zh-cn，改成 language=zh-cn

[设置]

→

[设置]

文本=zh-CN

language=zh-CN

3. remove-ini-key

作用：从 INI 文件中删除指定区段下的某个键值对。

用法：

remove-ini-key <INI 文件路径> <区段名> <目标键名>

remove-ini-key '/config/ultrahand/config.ini' 设置 语言

把语言=zh-cn 这一段删除

4. remove-ini-section

作用：从 INI 文件中删除指定的整个区段（包含区段下所有键值对）。

用法：

remove-ini-section <INI 文件路径> <区段名>

remove-ini-section '/bootloader/hekate_ipl.ini' '正版系统'

即把 hekate_ipl.ini 文件中，关于正版系统的一整段完全删除。

5. set-ini-val-matching-key

作用：编辑 INI 文件，为所有包含匹配模式键的区段，统一设置指定键的对应值。

用法：

set-ini-val-matching-key <待编辑 INI 文件路径> <匹配模式键> <目标键> <目标值>

set-ini-val-matching-key /config/ultrahand/config.ini mode spacing 4

表示在所有包含“mode”键的区段下，设置“spacing”键的值为“4”

6. remove-ini-key-matching-key

作用：编辑 INI 文件，从所有包含匹配模式键的区段中，删除指定的键值对。

用法：

remove-ini-key-matching-key <待编辑 INI 文件路径> <匹配模式键> <目标键名>

remove-ini-key-matching-key /config/ultrahand/config.ini table background

表示从所有包含“table”键的区段中，删除“background”键值对

源占位符函数

1. *

作用：常用作通配符或占位符，表示在执行过程中会动态填充的值，通常用于需要在运行时确定具体值的场景。

2. package_source

作用：允许用户从一个 Ultrahand 软件包的 INI 文件，跳转（转发）到另一个软件包的 INI 文件，帮助用户组织代码并串联更长的菜单列表。

用法：需配合 ;mode=forwarder 使用

例子：

[*作弊设置]

;mode=forwarder

package_source './cheat/package.ini'

注意，此处的目标 ini 文件可以是任何文本.ini'

3. list_source

作用：一种在 package.ini 文件中为下拉菜单存储变量的方式，作为列表源变量中单个值的占位符，代码会将 list_source 引用替换为对应的值。

例子：

```
list_source '(256, 512, 1024, 2048, 4096, 8192, 12288, 16384, 20480)'
```

4. list_file_source

作用：与 list_source 类似，但以文本文件的条目作为输入，代码会读取该文本文件的每行条目作为下拉菜单或命令的参数来源。

例子：

```
[*选择删除哪个系统]
```

```
list_file_source './name.txt' ( 拉取 name.txt 文件内每行的内容作为选项 )
```

```
remove-ini-section /bootloader/hekate_ipl.ini {list_file_source(*)}
```

5. file_source

作用：表示一个源文件或资源，作为实际文件名或源位置的占位符，代码在执行特定命令时会将 file_source 替换为对应的文件路径。

额外可访问变量：

- file_name：单个文件的名称；
- folder_name：文件所在父文件夹的名称；
- 若在 on: 区段中使用，意为“当指定源启用（on）时，执行对应命令”；
- 若在 off: 区段中使用，意为“当指定源禁用（off）时，执行对应命令”；
- 可判断开关状态或条件，进而选择执行对应的命令。

例子：

```
[*选择虚拟系统]
```

```
file_source '/emuMMC/*/' ( 拉取 emuMMC 文件夹下的所有子文件夹作为列表 )
```

```
'{file_name}' ( 自动变为你选定的文件或文件夹的名称。 )
```

```
'{folder_name}' ( 自动变为你选定的文件或文件夹的父文件夹名 )
```

6. ini_file_source

作用：与 list_file_source 类似，但以 INI 文件中的区段名称作为列表值的来源，同时保留与 ini_file 类似的其他 INI 文件操作功能。

例子：

```
[高级重启]
```

```
[*定向重启]
```

```
ini_file_source /bootloader/hekate_ipl.ini
```

```
filter config
```

reboot boot '{ini_file_source(*)}' (星号表示替换为你选定的值)

7. json_file_source

作用：用于定义 JSON 数据源，指定存储数据的 JSON 文件路径，代码会读取该 JSON 文件中的数据，并根据代码逻辑和场景执行操作（如显示数据、处理配置等）；

可额外指定 **jsonKey** 参数，用于获取 JSON 数据中特定键对应的内容。

额外可访问变量：

- jsonPath：JSON 数据源的文件路径；
- jsonKey：可选参数，用于获取 JSON 数据中特定键的值（具体实现可参考 Easy Installer 软件包示例）。

例子：

[*插件列表版本修改]

;mode=option

json_file_source ./Choose.json name (拉取指定路径的 json 文件中的 name 值作为列表)

unzip {json_file_source(*,path)} / (解压指定的值中的 path“地址”中指定的路径里面的文件，到根目录)

set-footer {json_file_source(*,name)} (将指定的值的 name 输出为页脚)

8. 以上代码，均可将_source 删除。直接将文件所有内容用于取值。

例子：

ini_file

[data]

ini_file '/bootloader/hekate_ipl.ini' (拉取 ini 文件内容中的“节名”作为列表)

set-ini-val './config.ini' hekate num 0

set-ini-val './config.ini' {ini_file(1)} num 1 (ini_file (数字) 选定从上往下数第几项的值) 此处第一项为 0，第二项为 1，以此类推！

set-ini-val './config.ini' {ini_file(2)} num 2

set-ini-val './config.ini' {ini_file(3)} num 3

json_file

拉取 json 文件为列表源，{json_file(索引数字,键名)}输出对应结果。注，索引数字以 0 开始。

[del]

json_file_source './nro.json'

del /config/ultra/{json_file_source(0,label)}

set-footer {json_file_source(0,name)}

9. 也可以直接用最前面的单词，然后双'内置可选项目。

例子：

```
json '["name":"4G 内存状态-默认 ", "label":"4G 内存状态"], {"name":"8G 内存状态 - 非硬改  
8G 勿动", "label":"8G 内存状态"}]
```

占位符函数

(ultrahand 内置的全局变量)

1. {title_id}

作用：读取并返回已打开的游戏的 TID

示例：'游戏 ID'='{title_id}'

显示为游戏 ID=0100450344248545

2. {build_id}

作用：读取已打开游戏的 bid

示例：'游戏 bid'='{build_id}'

返回十六位游戏的 bid 值

3. {local_ip}

作用：返回用户的 IP（本地局域网 IP）

4. {ram_vendor}

作用：返回设备内存（RAM）的厂商名称

示例：若设备使用三星内存，返回“Samsung”。

5. {ram_model}

作用：返回设备内存（RAM）的型号

示例：返回“M471A1K43BB1-CTD”（某型号内存的型号标识）。

6. {ams_version}

作用：返回设备上安装的 Atmosphere 的版本号

示例：若安装的是 1.7.1 版本，返回“1.7.1”。

7. {hos_version}

作用：返回设备上 Horizon OS（任天堂 Switch 官方系统）的版本号

示例：若系统版本为 18.1.0，返回“18.1.0”。

8. {cpu_speedo}

作用：返回设备 CPU 的 speedo 值（用于表示 CPU 性能等级的参数）

示例：返回“1600”（某设备的 CPU speedo 值）。

9. {cpu_iddq}

作用：返回设备 CPU 的 IDDQ 值（用于表示 CPU 电流特性的参数）

示例：返回“60”（某设备的 CPU IDDQ 值）。

10. {gpu_speedo}

作用：返回设备 GPU 的 speedo 值（用于表示 GPU 性能等级的参数）

示例：返回“1600”（某设备的 GPU speedo 值）。

11. {gpu_iddq}

作用：返回设备 GPU 的 IDDQ 值（用于表示 GPU 电流特性的参数）

示例：返回“60”（某设备的 GPU IDDQ 值）。

12. {soc_speedo}

作用：返回设备 SOC（系统级芯片）的 speedo 值（用于表示 SOC 性能等级的参数）

示例：返回“1600”（某设备的 SOC speedo 值）。

13. {soc_iddq}

作用：返回设备 SOC（系统级芯片）的 IDDQ 值（用于表示 SOC 电流特性的参数）

示例：返回“60”（某设备的 SOC IDDQ 值）。

14. {math(公式,ture)}

作用：进行+、-、*、/和()操作。 ,ture 为可选项，效果为强制输出整数结果

示例：{math(1.1+2.2)}。返回 3.3

{math(1.1+2.2,ture)}。返回 3

15. {slice(<字符串>,<起始顺序数字>,<结束顺序数字>)}

作用：根据提供的起始索引和结束索引，对字符串进行切片（提取指定区间的子字符串）

参数：<字符串>（待切片的原始字符串）、<起始索引>（切片的起始位置，从 0 开始计数）、<结束索引>（切片的结束位置，不包含该索引对应的字符）

示例：{slice(Test String, 0, 4)} 返回'Test'

14. {decimal_to_hex(<十进制数>)}

作用：将十进制数转换为十六进制字符串（大写）

参数：<十进制数>（待转换的十进制整数）

示例：{decimal_to_hex(255)} 返回“FF”

15. {ascii_to_hex(<ASCII 字符串>)}

作用：将 ASCII 字符串转换为对应的十六进制表示（每个字符对应两位十六进制数）

参数：<ASCII 字符串>（待转换的 ASCII 字符串）

示例：{ascii_to_hex("Test")} 返回“54657374”（“T”对应“54”、“e”对应“65”、“s”对应“73”、“t”对应“74”）

16. {hex_to_rhex(<十六进制字符串>)}

作用：将十六进制字符串转换为反向字节序的十六进制字符串（每两位为一个字节反向）

参数：<十六进制字符串>（待转换的十六进制字符串）

示例：{hex_to_rhex("12345678")} 返回“78563412”

17. {hex_to_decimal(<十六进制字符串>)}

作用：将十六进制字符串转换为对应的十进制数

参数：<十六进制字符串>（待转换的十六进制字符串，不区分大小写）

示例：{hex_to_decimal("FF")} 返回“255”

18. {split(<字符串>,<分隔模式>,<顺序数字>)}

作用：根据指定的分隔模式将字符串分割为多个部分，并返回指定索引对应的部分（从 0 开始计数）

参数：<字符串>（待分割的原始字符串）、<分隔模式>（用于分割字符串的字符或字符串，如“-”）、<索引>（需返回的分割部分的位置）

示例：{split('网址-1234',-,0)}返回“网址”

19. {value}

作用：返回滑动条（trackbar）当前选中的值

示例：在滑动条控件中使用，用于显示当前调整的数值（如亮度值、音量值）。

20. {index}

作用：返回当前选择项或列表中选中项的顺序数字（从 0 开始计数）

示例：用于标识下拉菜单或列表中选中项的位置。

21. {list_source(索引|数字)}

作用：从之前用 list_source 定义的列表中，获取指定顺序数字对应的条目

参数：顺序数字（列表中条目的位置，从 0 开始计数）

示例：{list_source(0)} 返回列表中的第一个条目

22. {json_source(*,<键名>)}

作用：从 JSON 数据源中，获取指定键名对应的数值或字符串

参数：<键名>（JSON 数据中待查询的键）

示例：{json_source(title)} 返回 JSON 数据中“title”键对应的内容

23. {ini_file(<区段>, <键名>)}

作用：从 INI 文件的指定区段中，获取指定键名对应的数值或字符串

参数：<区段>（INI 文件中的目标区段，如“Settings”）、<键名>（该区段下的目标键）

示例：{ini_file("Settings", "Language")} 返回 INI 文件“Settings”区段下“Language”键的值 'zh-cn'

```
[settings]
```

```
Language=zh-cn
```

24. {json_file(索引|数字,键名)}

作用：从 json_file 拉取的列表中取值，索引从 0 开始计算

指定命令：

1.

;footer=

默认页脚（选项后面的标注），使该命令在未操作的情况下，在后面显示这段后面输入的内容。

;footer_highlight= true / false

页脚高亮开关，填入 true 时，页脚会高亮显示。填入 false 时，页脚颜色和普通颜色一样。

2.

;selection_mini=true

更窄的边框

3.

;hold=true

长按 A 键才执行命令，而不是点击马上执行。

例子：

[长按以关机]

```
;hold=true
```

```
shutdown
```

4.

`:system=`

显示命令。后面填入 mariko 或者 erista。

只在对应型号的 switch 中会显示该行命令。

5. 2.2.8 更新的命令行

`;state=`

显示命令。后附 **handheld**(手持)或 **docked**(底座)

只在对应的模式下显示该命令

命令模式：

1.

`;mode=default`

作用：默认命令模式，按命令原始形式执行，无特殊交互或行为，无需显式指定其他模式时使用。

示例：

[重启]

`;mode=default`

reboot

2.

`;mode=toggle`

作用：开关模式命令，在“开启（on）”和“关闭（off）”间切换，执行时根据当前状态执行对应命令并切换状态，适用于可启用/禁用的设置。

如果要定义数据源，需要在**`;mode=toggle`** 下方，而不是 on 或 off 下方。

参数：

toggle?on：设置开关默认状态为“开启”

toggle?off：设置开关默认状态为“关闭”

示例：

[背光亮度]

`;mode=toggle?on`

on:

backlight on

off:

backlight off

3.

;mode=option

作用：选项模式命令，显示预定义选项列表供用户选择，适用于需从多个配置/设置中挑选的场景。

示例：

[*频率设置]

;mode=option

list_source '(低,中,高)'

4.

;mode=trackbar

作用：滑动条模式命令，创建标准滑动条，允许用户在最小/最大数值范围内调整，适用于需调整数值的设置。

参数：

min_value：滑动条最小整数值

max_value：滑动条最大整数值

units：数值显示单位（如“%”“Hz”）

unlocked：可选，允许滑动条即时交互（无需确认）

on_every_tick：可选，滑动条每调整一次执行一次命令（默认关闭）

示例：

[亮度]

;mode=trackbar

min_value=0

max_value=100

units=%

backlight={value}

5.

;mode=step_trackbar

作用：步进滑动条模式命令，与普通 trackbar 类似，但需按固定步长调整数值，适用于按固定间隔调整的设置。

参数：

steps：滑动条总步数（步长 = $(\text{max_value} - \text{min_value}) / (\text{steps} - 1)$ ）

示例：

[档位调整]

```
;mode=step_trackbar
```

```
min_value=0
```

```
max_value=5
```

```
steps=6
```

6.

;mode=named_step_trackbar

作用：命名步进滑动条模式命令，步进滑动条的变种，每个步进取预定义名称（非纯数值），适用于需明确标签的步进设置。

参数：

list_source：为每个步进设置名称的源列表

示例：

[性能模式]

```
;mode=named_step_trackbar
```

```
list_source '(低, 中, 高, 极限)'
```

7.

;mode=slot

作用：插槽模式命令，允许用户从预定义选项组中执行命令，支持自定义选项标签和页脚说明，比 option 模式对显示元素控制更灵活，适用于分组选择场景。

示例：

[存档位置]

```
;mode=slot
```

```
list_source '(test a, test b, test c)'
```

```
set_footer list_source(*)
```

8.

;mode=forwarder

作用：转发模式命令，指向另一个软件包的.ini 文件，用户选择后跳转至目标软件包或执行其命令，适用于模块化串联命令/软件包。

参数：package_source：目标软件包.ini 文件的路径

示例：

[系统]

[*增加内存]

;mode=forwarder

package_source './Atmosphere/package.ini'

9.

;mode=table

作用：表格显示模式命令，以表格形式渲染结构化信息，支持定义行列、对齐方式等，适用于显示系统信息、设备参数等结构化数据。

参数：

1.

;alignment=left/center/right

作用：设置表格列内文本的对齐方式，可选 left（左对齐）、center（居中）、right（右对齐），用于规范表格内容的排版对齐效果。

示例：

[表格文本对齐设置]

;mode=table

alignment=center

header=true

2.

;header=true/false

作用：控制是否显示表格的标题行，true 为显示，false 为隐藏，适用于是否需要展示表格表头的场景。

示例：

[表格标题行设置]

;mode=table

header=false

spacing=4

3.

;spacing=数值

作用：设置表格行间距，数值为自定义的间距单位，调整表格行与行之间的空白距离，优化视觉间距。

示例：

[表格行间距设置]

;mode=table

spacing=4

alignment=left

4. gap

;gap=数值

作用：设置表格下方的空白间隔，数值为自定义的间隔单位，在表格与其他界面元素间预留空白，避免视觉拥挤。

示例：

[表格下方间隔设置]

;mode=table

gap=20

header=true

5.

;wrapping_mode=char

作用：设置文本换行模式为按字符换行，文本达到指定宽度时以单个字符为单位换行，适用于精准控制字符级换行的场景。

示例：

[文本换行设置]

;mode=table

wrapping_mode=char

alignment=left

header=true

6.

```
;wrapping_indent=true
```

作用：启用文本换行后的缩进功能，换行后首行文本按预设规则缩进，提升多行文本排版可读性。

示例：

[文本缩进设置]

```
;mode=table
```

```
wrapping_indent=true
```

```
spacing=4
```

```
gap=20
```

7.

```
;background=false
```

作用：关闭表格/文本区域的背景显示效果，仅渲染文本和边框（若有），追求简洁的界面视觉效果。

示例：

[界面背景设置]

```
;mode=table
```

```
background=false
```

```
header_indent=true
```

8.

```
;header_indent=true
```

作用：启用表格标题行的缩进功能，标题行首行按预设规则缩进，区分标题与内容，提升结构化展示效果。

示例：

[表格标题缩进设置]

```
;mode=table
```

```
header_indent=true
```

```
section_text_color=header
```

9.

```
;section_text_color=header
```

作用：将章节文本颜色设为与表格标题行相同的颜色，保持界面文本色彩风格统一，适用于分章节的结构化展示。

示例：

[章节文本配色设置]

```
;mode=table
```

```
section_text_color=header
```

```
wrapping_mode=char
```

示例：

[系统提示]

```
;mode=table
```

```
alignment=left
```

```
header=true
```

10.

```
;section_text_color 修改左侧颜色
```

```
;info_text_color 修改右侧的颜色
```

```
'左侧'='右侧'
```

十六进制编辑命令

1. hex-by-offset

作用：在文件的指定偏移量（offset）处，用提供的十六进制数据编辑文件内容。

用法：hex-by-offset <文件路径> <偏移量> <十六进制数据>

例子：若要在 D:\test.bin 文件的第 100 个字节位置（偏移量 100）写入十六进制数据 A3 B7，命令为 hex-by-offset D:\test.bin 100 A3 B7。

2. hex-by-custom-offset

作用：在文件中找到与指定自定义模式匹配的位置后，从该位置的偏移量处，用提供的十六进制数据编辑文件内容。

用法：hex-by-custom-offset <文件路径> <自定义模式> <偏移量> <十六进制数据>

- 例子：若在 D:\test.bin 中找到模式 5F 8C 的位置后，再从该位置往后偏移 5 个字节处写入 12 34，命令为 hex-by-custom-offset D:\test.bin 5F 8C 5 12 34。

3. hex-by-custom-decimal-offset

作用：在文件中找到与指定自定义模式匹配的位置后，从该位置的偏移量处，用提供的十进制

数据编辑文件内容（十进制数据会自动转换为十六进制）。

用法：hex-by-custom-decimal-offset <文件路径> <自定义模式> <偏移量> <十进制数据>

例子：若在 D:\test.bin 中找到模式 2A 6D 后，从该位置偏移 3 字节处写入十进制 255（自动转十六进制 FF），命令为 hex-by-custom-decimal-offset D:\test.bin 2A 6D 3 255。

4. hex-by-custom-rdecimal-offset

- 作用：在文件中找到与指定自定义模式匹配的位置后，从该位置的偏移量处，用提供的反向十进制数据编辑文件内容（十进制数据先转十六进制，再反向处理）。

用法：hex-by-custom-rdecimal-offset <文件路径> <自定义模式> <偏移量> <反向十进制数据>

例子：若在 D:\test.bin 中找到模式 78 90 后，从该位置偏移 2 字节处写入反向十进制 156（十进制 156 转十六进制 9C，反向为 C9），命令为 hex-by-custom-rdecimal-offset D:\test.bin 78 90 2 156。

5. hex-by-swap

作用：将文件中指定的十六进制数据替换为另一组十六进制数据，可选择指定替换第几个匹配项（不指定则替换所有）。

用法：hex-by-swap <文件路径> <待替换十六进制数据> <替换后十六进制数据> [匹配项索引]

例子：若要将 D:\test.bin 中所有 AB CD 替换为 EF

12，命令为 hex-by-swap D:\test.bin AB CD EF 12；若只替换第 3 个匹配项，命令为 hex-by-swap D:\test.bin AB CD EF 12 3。

6. hex-by-string

作用：将文件中指定的字符串数据替换为另一组字符串数据（字符串自动转对应十六进制后替换），可选择指定替换第几个匹配项。

用法：hex-by-string <文件路径> <待替换字符串> <替换后字符串> [匹配项索引]

例子：若要将 D:\test.txt 中所有字符串 "old"（转十六进制 6F 6C 64）替换为 "new"（转十六进制 6E 65 77），命令为 hex-by-string D:\test.txt old new。

7. hex-by-decimal

作用：将文件中指定的十进制数据替换为另一组十进制数据（十进制先转十六进制后替换），可选择指定替换第几个匹配项。

用法：hex-by-decimal <文件路径> <待替换十进制数据> <替换后十进制数据> [匹配项索引]

例子：若要将 D:\test.bin 中所有十进制 100 (转十六进制 64) 替换为十进制 200 (转十六进制 C8)，命令为 hex-by-decimal D:\test.bin 100 200。

8. hex-by-rdecimal

作用：将文件中指定的反向十进制数据替换为另一组反向十进制数据（十进制先转十六进制，再反向处理后替换），可选择指定替换第几个匹配项。

用法：hex-by-rdecimal <文件路径> <待替换反向十进制数据> <替换后反向十进制数据> [匹配项索引]

例子：若要将 D:\test.bin 中所有反向十进制 123 (转十六进制 7B，反向为 B7) 替换为反向十进制 456 (转十六进制 1C8，反向为 8C1)，命令为 hex-by-rdecimal D:\test.bin 123 456。

9. hex-by-pattern

作用：将文件中与指定十六进制模式匹配的数据，替换为提供的十六进制数据。

用法：hex-by-pattern <文件路径> <待匹配十六进制模式> <替换后十六进制数据>

例子：若要将 D:\test.bin 中所有匹配模式 ?? 3A (?? 为任意十六进制数) 的数据替换为 4F 3A，命令为 hex-by-pattern D:\test.bin ?? 3A 4F 3A。

模组转换命令：

1. pchtxt2ips：

将 .pchtxt 格式的模组文件转换为 .ips 格式的补丁文件。

使用方法：pchtxt2ips <pchtxt 文件路径> <输出文件路径>

2. pchtxt2cheat

将单个 .pchtxt 模组转换并安装为可直接使用的作弊码。

使用方法：pchtxt2cheat <pchtxt 文件路径>