

# 加密工具使用说明

---

## 环境准备

---

安装 Python 2.7.x 版本。（工具基于 2.7.10 版本开发）

## 命令行使用说明

---

```
python ResEncrypt.py -h
usage: ResEncrypt [-h] -s SRC [-d DST] [--rm-src] [--copy] [--cfg
EXCLUDE_CFG]
```

对资源文件进行加密和压缩

optional arguments:

-h, --help	show this help message and exit
-s SRC, --src SRC	指定源文件或者文件夹。
-d DST, --dst DST	指定目标路径。默认与源文件位置相同。
--rm-src	删除原始的文件
--copy	指定拷贝非加密文件
--cfg EXCLUDE_CFG	指定一个配置文件，这个文件用于配置排除某些资源文件

参数的具体说明如下表：

参数	可用值	是否必须	说明
-h, --help	-	否	显示帮助信息
-s, --src	path/to/resourcefiles	是	指定源文件或者文件夹。可以是绝对路径，或者相对于当前目录的相对路径。
-d, --dst	path/to/output/dir	否	指定输出文件夹路径。可以是绝对路径，或者相对于当前目录的相对路径。如果不指定，那么加密文件将保存在原始资源文件的同级文件夹下。
--rm-src	-	否	如果使用此参数，原始的文件将在加密后删除。
--copy	-	否	如果使用此参数，那么不需要加密的文件也会被拷贝到目标目录下
--cfg	path/to/exclude-cfg/file	否	指定一个配置文件路径。这个文件中配置哪些 lua 文件不会被加密。可以是绝对路径，或者相对于当前目录的相对路径。

备注：

- --copy 参数只在 -s 参数为文件夹，目标路径与源路径不同时生效。
- --cfg 参数指定的配置文件需要为 json 文件，格式如下：

```
[
    "exclude-dir/",
    "exclude/path/to/the.lua",
    "*cfg_*.lua"
]
```

- 排除文件的实现逻辑：
  1. 将每条规则字符串中的 `.` 转换为 `\.`，`*` 转换为 `.*`
  2. 对于每个 lua 文件，获取相对于源路径的相对路径。
  3. 将相对路径按照正则表达式的方式匹配每条规则。
  4. 如果某一条规则匹配成功，那么文件将被排除；否则，文件不会被排除

## 示例

- `python ResEncrypt.py -s ./src -d ./src_output`

将 src 文件夹下的文件加密后存储到 src\_output 文件夹中

- `python ResEncrypt.py -s ./src --cfg ./exclude-cfg.json --rm-src`

对 src 文件夹下的文件进行加密并删除原始的文件。加密过程中应用 exclude-cfg.json 中的排除规则

- `python ResEncrypt.py -s ./src -d ./src_output --copy`

将 src 文件夹下的文件加密后存储到 src\_output 文件夹中，并且未加密的文件也拷贝到 src\_output 文件夹中。