

Q2`25-Slots-图片资源压缩-程序

现状:

1. 发布阶段, web 端资源做 webp 压缩; 【生效中】
2. 研发阶段, 美术在提交资源时视情况自行操作图片压缩; 【已废弃】
该流程已于多年前废弃, 目前 native 端图片资源未做任何压缩处理;
3. 资源数量日趋庞大, 下载、运行负担日重;

需求:

1. 支持一键、选择性压缩;
2. 不改变原始资源, 避免增加美术修改负担;
3. 避免增加发版过程, 目前发版时长已经很长了;
4. 避免每次发布资源都要等待压缩, 压缩过程较为耗时;
5. 多端同步;

实现方案:

1. “发布资源到工程”脚本, 增加图片压缩过程: (`PlugIns/resourcePublish.py`)
 - a. 仍采用 tinyPNG (熊猫) 压缩;
 - b. 默认对所有发布的图片资源做压缩处理;
 - c. 排除默认、指定的忽略文件;

```
1 def publishImageFile(srcPathFileName, dstPathFileName,
2   hdDstPathFileName):
3     _, fileName = os.path.splitext(srcPathFileName)
4
5     if fileName == "ipad.png" or fileName == "iphone.png":
6         return;
```

```

7
8         if srcPathFileName.endswith(".png") or
srcPathFileName.endswith(".jpg"):
9             dstPathFileName = dstPathFileName.replace("_hd.", ".")
10            if not os.path.exists(os.path.dirname(dstPathFileName)):
11                os.makedirs(os.path.dirname(dstPathFileName))
12
13            shutil.copyfile(srcPathFileName, dstPathFileName);
14
15            if(srcPathFileName.endswith("_hd.png") or
srcPathFileName.endswith("_hd.jpg")):
16
17                if hdDstPathFileName:
18
19                    if not
os.path.exists(os.path.dirname(hdDstPathFileName)):
20                        os.makedirs(os.path.dirname(hdDstPathFileName))
21                        shutil.copyfile(srcPathFileName, hdDstPathFileName);
22
23                if not pkgutil.find_loader("PIL"):
24                    print "请输入登录密码安装插件需要的依赖库:"
25                    os.system("sudo pip install Pillow")
26
27                from PIL import Image
28
29                hdImage = Image.open(dstPathFileName)
30                (w, h) = hdImage.size
31                sdImage = hdImage.resize((int(w*0.5),
int(h*0.5)), Image.ANTIALIAS)
32                hdImage.close()
33                sdImage.save(dstPathFileName)
34                sdImage.close()
35                # 压缩文件
36                if not ignore_util.in_png_compress_ignore(srcPathFileName):
37                    if not md5_util.isMD5Equal(srcPathFileName):
38                        if not (dstPathFileName.endswith("_hd.png") or
dstPathFileName.endswith("_hd.jpg")):
39                            print "Compress Image File:%s"%(dstPathFileName)
40                            os.system("python ./compressImage.py %s%"
(dstPathFileName))
41                else:
42                    print "File(%s)is Not Image File"%(srcPathFileName)

```

2. 增加发布文件 md5 值记录, 防止重复发布、压缩: (Plugins/md5_util.py)

已发布记录文件: __check__/published.md5 【及时提交、同步】

- a. 记录发布过的文件的 md5 值;

- b. 文件无变更（记录中的 md5 值与本地文件 md5 值一致）时跳过发布；
- c. 遇到特殊情况，可以删掉 `published.md5` 文件，重新发布；

```
1  #!/usr/bin/env python
2  # coding: utf-8
3
4  import os
5  import sys
6
7  # 获取 srcPathFileName 在 Resources 路径下的相对路径
8  def get_src_relative_path(srcPathFileName):
9      relativePath = srcPathFileName.split("Resources")[1]
10     return relativePath.lstrip("/")
11
12 # 获取 md5 记录文件路径
13 def get_md5_record_path(srcPathFileName):
14     resourcesPath = srcPathFileName.split("Resources")[0] + "Resources"
15     md5TxtPath = resourcesPath + "/../__check__/published.md5"
16     return md5TxtPath
17
18 # 获取文件 md5 值
19 def get_file_md5(srcPathFileName):
20     md5Value = os.popen('md5sum "%s"'%(srcPathFileName)).read().strip()
21     md5Value = md5Value.split(" ")[0]
22     return md5Value
23
24 # 记录 md5 值
25 def recordMd5(srcPathFileName):
26     md5TxtPath = get_md5_record_path(srcPathFileName)
27     # 如果 md5.txt 路径下不存在，就创建一个
28     if not os.path.exists(os.path.dirname(md5TxtPath)):
29         os.makedirs(os.path.dirname(md5TxtPath))
30     if not os.path.exists(md5TxtPath):
31         os.system('touch "%s"'%(md5TxtPath))
32
33     relativePath = get_src_relative_path(srcPathFileName)
34     md5Value = get_file_md5(srcPathFileName)
35     # 读取 md5 记录
36     md5TxtContent = ""
37     md5TxtContent = os.popen('cat "%s"'%(md5TxtPath)).read().strip()
38     # 判断 md5.txt 路径下是否已经存在 srcPathFileName 的 md5 值
39     if md5TxtContent.find(relativePath) == -1:
40         # 如果不存在，就将 srcPathFileName 的 md5 值写入 md5.txt 路径下
41         with open(md5TxtPath, 'a') as f:
42             f.write(md5Value + " " + relativePath + "\n")
43         f.close()
```

```

44     else:
45         # 如果存在, 就将 srcPathFileName 的 md5 值替换掉 md5.txt 路径下的旧
        值, 并保存
46         with open(md5TxtPath, 'r') as f:
47             content = f.read()
48             content = content.replace(md5Value + " " + relativePath +
        "\n", md5Value + " " + relativePath + "\n")
49             f.close()
50         with open(md5TxtPath, 'w') as f:
51             f.write(content)
52             f.close()
53
54     def isMD5Equal(srcPathFileName):
55         md5TxtPath = get_md5_record_path(srcPathFileName)
56         # md5TxtPath 不存在直接返回 True
57         if not os.path.exists(md5TxtPath):
58             return False
59         relativePath = get_src_relative_path(srcPathFileName)
60         md5TxtContent = os.popen('cat "%s"%(md5TxtPath)').read().strip()
61         if md5TxtContent.find(relativePath) == -1:
62             return False
63         else:
64             # 如果存在, 就将 srcPathFileName 对应的 md5 值读取出来
65             with open(md5TxtPath, 'r') as f:
66                 content = f.read()
67                 md5Value = content.split(relativePath)[0].split("\n")
        [-1].strip()
68                 f.close()
69             # 获取 srcPathFileName 的 md5 值
70             new_md5Value = get_file_md5(srcPathFileName)
71             # 判断 md5.txt 路径下的 md5 值是否与 srcPathFileName 的 md5 值一致
72             if new_md5Value == md5Value:
73                 return True
74             else:
75                 return False
76
77
78     def remove_from_published_md5(srcPathFileName):
79         record_file_path = get_md5_record_path(srcPathFileName)
80         if not os.path.exists(record_file_path):
81             return
82         relativePath = get_src_relative_path(srcPathFileName)
83         print("remove_from_published_md5:%s"%(relativePath))
84         with open(record_file_path, 'r') as f:
85             record = f.read().splitlines()
86             print("record:%s"%(record))
87             f.close()
88         # 遍历 record, 移除 relativePath 对应的记录, 并保存到文件中

```

```

89     for i in range(len(record)):
90         if record[i].find(relativePath) != -1:
91             record.pop(i)
92             break
93     with open(record_file_path, 'w') as f:
94         for i in record:
95             f.write(i + '\n')
96     f.close()
97
98 if __name__ == "__main__":
99     if len(sys.argv) < 3:
100         print('Usage: python md5_util.py operation srcPathFileName
[param1 param2 ...]')
101         print('operation: [record,remove_from_published_md5]')
102         exit(1)
103
104     # 打印参数
105     print('operation:', sys.argv[1])
106     print('srcPathFileName:', sys.argv[2])
107     print('param:', sys.argv[3:])
108     # 根据 operation 执行不同的操作
109     operation = sys.argv[1]
110     if operation == 'record':
111         recordMd5(sys.argv[2])
112     elif operation == 'remove_from_published_md5':
113         remove_from_published_md5(sys.argv[2])
114
115     exit(0)

```

3. 增加忽略压缩功能: (PlugIns/ignore_util.py)

忽略记录文件: `__check__/png_compress_ignore.txt` 【及时提交、同步】

- 默认忽略列表: `[/symbol/,"common/","slot/"]`;
- 从 `png_compress_ignore.txt` 中获取忽略列表;
- 忽略列表做 `==`、包含匹配;

```

1  #!/usr/bin/env python
2  # coding: utf-8
3
4  import os
5  import sys
6
7  png_compress_common_ignore_list = ["/symbol/","common/","slot/"]
8
9  # 获取 srcPathFileName 在 Resources 路径下的相对路径

```

```

10 def get_src_relative_path(srcPathFileName):
11     relativePath = srcPathFileName.split("Resources")[1]
12     return relativePath.lstrip("/")
13
14 # 获取图片压缩忽略文件路径
15 def get_png_compress_ignore_path(srcPathFileName):
16     resourcesPath = srcPathFileName.split("Resources")[0] + "Resources"
17     md5TxtPath = resourcesPath + "/../__check__/png_compress_ignore.txt"
18     return md5TxtPath
19
20 def get_png_compress_ignore(srcPathFileName):
21     ignore_file_path = get_png_compress_ignore_path(srcPathFileName)
22     if os.path.exists(ignore_file_path):
23         with open(ignore_file_path, 'r') as f:
24             ignore = f.read().splitlines()
25             f.close()
26             return ignore
27     else:
28         return []
29
30 def in_png_compress_ignore(srcPathFileName):
31     ignores = get_png_compress_ignore(srcPathFileName)
32     srcPathFileName = srcPathFileName[srcPathFileName.find('Resources/')+10:]
33     if srcPathFileName in ignores:
34         return True
35     for ignore in png_compress_common_ignore_list:
36         if ignore in srcPathFileName:
37             return True
38     for ignore in ignores:
39         if ignore in srcPathFileName:
40             return True
41     return False
42
43 def add_to_png_compress_ignore(srcPathFileName):
44     ignore_file_path = get_png_compress_ignore_path(srcPathFileName)
45     # 如果 ignore_file_path 路径下不存在, 就创建一个
46     if not os.path.exists(os.path.dirname(ignore_file_path)):
47         os.makedirs(os.path.dirname(ignore_file_path))
48     if not os.path.exists(ignore_file_path):
49         os.system('touch "%s"'%(ignore_file_path))
50
51     relativePath = get_src_relative_path(srcPathFileName)
52     # 添加新的条目前, 先检查是否已经存在
53     with open(ignore_file_path, 'r') as f:
54         ignore = f.read().splitlines()
55         if relativePath in ignore:
56             f.close()

```

```

57         return
58     with open(ignore_file_path, 'a') as f:
59         f.write(relativePath + '\n')
60         f.close()
61
62
63 def remove_from_png_compress_ignore(srcPathFileName):
64     ignore_file_path = get_png_compress_ignore_path(srcPathFileName)
65     if not os.path.exists(ignore_file_path):
66         return
67
68     relativePath = get_src_relative_path(srcPathFileName)
69     with open(ignore_file_path, 'r') as f:
70         ignore = f.read().splitlines()
71         if relativePath in ignore:
72             ignore.remove(relativePath)
73         f.close()
74     with open(ignore_file_path, 'w') as f:
75         for i in ignore:
76             f.write(i + '\n')
77         f.close()
78
79 if __name__ == "__main__":
80     if len(sys.argv) < 3:
81         print('Usage: python ignore_util.py operation srcPathFileName
82         [param1 param2 ...]')
83         print('operation:
84         [add_to_png_compress_ignore,remove_from_png_compress_ignore]')
85         exit(1)
86
87     # 打印参数
88     # print('operation:', sys.argv[1])
89     # print('srcPathFileName:', sys.argv[2])
90     # print('param:', sys.argv[3:])
91
92     # 根据 operation 执行不同的操作
93     operation = sys.argv[1]
94     if operation == 'add_to_png_compress_ignore':
95         add_to_png_compress_ignore(sys.argv[2])
96     elif operation == 'remove_from_png_compress_ignore':
97         remove_from_png_compress_ignore(sys.argv[2])
98
99     exit(0)

```

4. 标记颜色功能，绑定压缩忽略功能：（ Plugin/setTag.sh ）

忽略记录文件： `__check__/tag.txt` 【及时提交、同步】

- a. 添加绿色标记时，记录到 `png_compress_ignore.txt`、`tag.txt` ；

- b. 移除绿色标记时, 移除 png_compress_ignore.txt、tag.txt 中的记录;
- c. 刷新标记时, 从 tag.txt 获取标记记录, 逐条应用“添加标记”操作;

```
1  #!/bin/sh
2  # title: 标记颜色
3  # filters: subfolder file
4  # options: global wait
5  # icon: none
6  # order: 190
7  # submenu: 2 红色 red
8  # submenu: 1 橙色 orange
9  # submenu: 3 黄色 yellow
10 # submenu: 6 绿色 green
11 # submenu: 4 蓝色 blue
12 # submenu: 5 紫色 purple
13 # submenu: 7 灰色 gray
14 # submenu: 0 清除 none
15 # submenu: 8 刷新标记
16
17 col="0"
18 if [ $4 ]; then
19     col=$4
20 fi
21
22 process_color_tag() {
23     local col=$1
24     local file_path=$2
25
26     script_dir=$(dirname "$0")
27     pushd "$script_dir"
28     if [ "$col" = "6" ]; then
29         python ignore_util.py add_to_png_compress_ignore "$file_path"
30     elif [ "$col" = "0" ]; then
31         python ignore_util.py remove_from_png_compress_ignore
"$file_path"
32         python md5_util.py remove_from_published_md5 "$file_path"
33     fi
34     popd
35 }
36
37 # 将$col存储在本地文件中, 存储文件路径为: $1 中与/Resources/同级的目录,
__check__/下的tag.txt
38 # 存储格式为: $1 在 Resources/下的相对路: $col
39 # 获取$1在Resources/下的相对路径
40 relative_path=$(echo $1 | sed 's/.*\/Resources\\///')
41 # 获取 Resources/的父目录
```



```

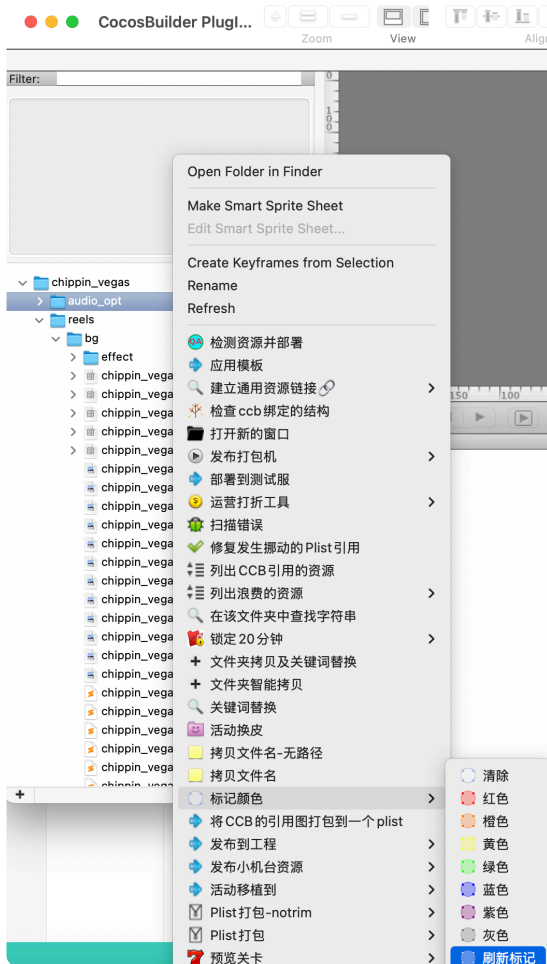
42 root_path=$(echo $1 | sed 's/\/Resources\/.*//')
43 # 获取 Resources/的父目录下的__check__/tag.txt
44 tag_path=$root_path/__check__/tag.txt
45 # 如果tag.txt不存在, 则创建
46 if [ ! -f $tag_path ]; then
47     mkdir -p $root_path/__check__
48     touch $tag_path
49 fi
50
51 if [ $col = "8" ]; then
52     echo 正在刷新标记
53     # 读取tag.txt中的内容
54     while read line; do
55         # 获取$1在Resources/下的相对路径
56         relative_path=$(echo $line | awk -F ':' '{print $1}')
57         # 获取$1在Resources/下的相对路径
58         col=$(echo $line | awk -F ':' '{print $2}')
59         target_path=$root_path/Resources/$relative_path
60         # 如果$1不存在, 则跳过
61         if [ ! -f $target_path ]; then
62             continue
63         fi
64         echo 正在标记颜色 $col
65         process_color_tag $col $target_path
66         osascript -e "tell application \"Finder\" to set label index of
alias POSIX file \"$target_path\" to $col"
67     done < $tag_path
68     exit 0
69 fi
70
71 if [ $col = "0" ]; then
72     sed -i '' "\|$relative_path:d" "$tag_path"
73 else
74     if grep -q "$relative_path:" $tag_path; then
75         sed -i '' "\|$relative_path:d" "$tag_path"
76     fi
77     echo $relative_path:$col >> $tag_path
78 fi
79 process_color_tag $col $1
80 osascript -e "tell application \"Finder\" to set label index of alias
POSIX file \"$1\" to $col"

```

操作说明:

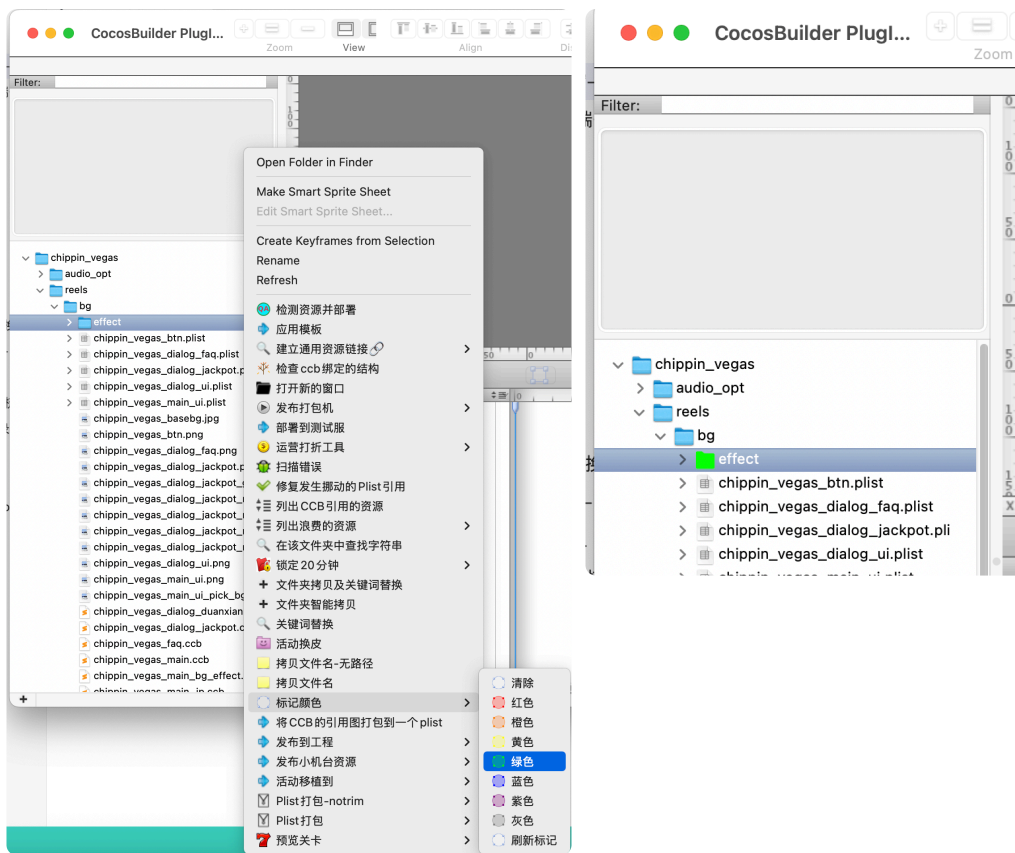
1. 刷新标记:

- 拉取 git 以同步记录文件;
- 选择工程中任意目录、文件, 右键 -> 标记颜色 -> 刷新标记;
- 聚焦回 cocosbuilder (点击工程中任意文件、文件夹), 查看标记状态;



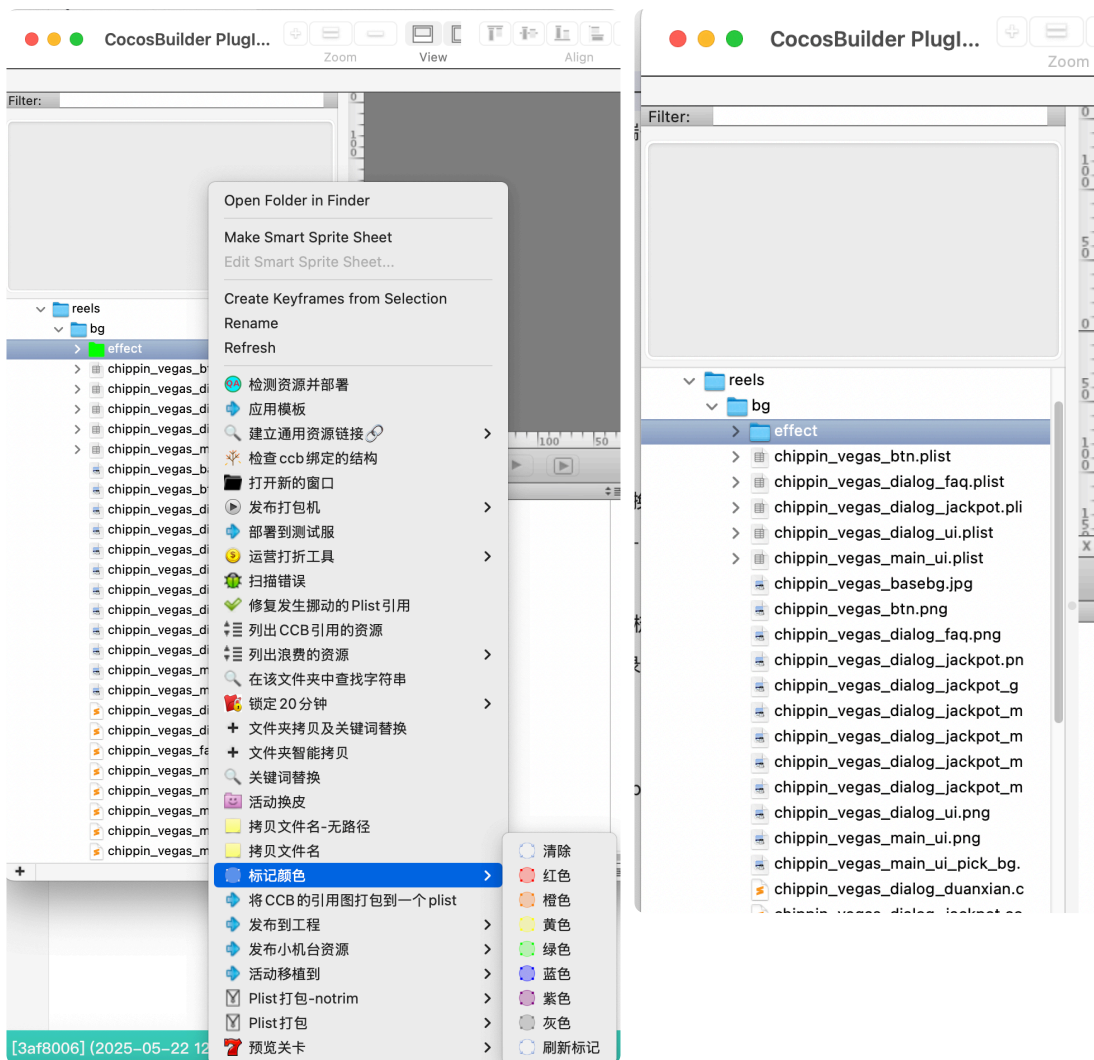
2. 增加标记: 绿色标记代表“不压缩”

- 可针对目录、或文件;
- 聚焦回 cocosbuilder (点击 cocosbuilder 任意区域), 查看标记状态;



3. 移除标记:

- 选中已有颜色的文件、目录，右键 -> 标记颜色 -> 清除;
- 移除绿色标记后，表示该资源发布时会被压缩;



! 注意:

1. 提交 `__check__` 目录及其下所有文件:
所有需要同步的记录、标记都在这个目录下, 有变更必须及时提交;
2. 需要新增、移除标记前: 拉取 git -> 刷新标记;
不操作刷新标记无法同步颜色显示状态, 但不影响已经有过的标记记录生效 (只是看不到颜色而已);
3. (程序) 首次大面积资源发布, 会多耗费 1-2 分钟时间逐个处理压缩;
4. (程序) 首次发布后, 请及时提交发布后的资源和 `__check__` 目录;
5. (程序) 如果出现原因导致的发布后资源不符合预期的情况, 可以手动删除 `__check__/published.md5` 文件, 重新发布;

