

mergeWtHtFile.sh说明

根据测风塔层高和测风塔编号分拣数据的程序

软件当前版本: v20.01.000

文档更新时间: 2023-04-15

作者 : fu.sky

版本变化历史记录:

2023-04-15 : 初始版本 : v20.01.000

目录

- i. -----[功能说明](#)
- ii. -----[软件部署](#)
- iv. -----[软件目录结构说明](#)
- vii. -----[附配置文件样例](#)

功能说明

简述: 根据特殊格式的测风塔源数据文件,按测风塔编号和层高分类生成结果文件

使用此脚本的条件

1. 在脚本的同级目录下有配置文件cfg/cfg.cfg, 且配置文件已经按要求配置完成

2. 源文件需要满足如下条件:

- (1) 文件名最好满足格式: JSDLFD_20110614_1930_CFT.WPD, 即
"前缀_时间_后缀"
- (2) 文件内容同一个文件中包含:

同一个时刻所有层高采集的数据,

且文件第一行内容就为数据时刻, 格式为:// 2011-06-14_19:30:00
(格式解释: 顶格//空格yyy-mm-dd_hh24:mi:ss)

层高数据类似如下:

```
<MastData::0001>
@id    属性项      数值
#1      WS_10      9.1
#2      WS_BZC_10   3
#3      WS_SS_10    9.1
#4      WS_MAX_10    3
#5      WD_10       9.1
#6      WD_SS_10     3
</MastData::0001>
```

其中属性项中"_线后的数字"表示层高;
MastData后的数字代表测风塔编号
(同一个文件中可以有多个测风塔编号包含的内容)

3. 结果文件说明:

文件名类似: cft_001_010_20230307-20230307.csv

文件名规则: 前缀_测风塔编号_数据层高_开始时间-结束时间_文件后缀
其中开始时间和结束时间精确到天(因此开始与结束是同一个值)

文件名只有: "前缀 后缀"在配置文件中可配置

文件内容: 将同一个测风塔同一个层高的数据, 当天所有时刻的数据合成一起
数据类似如下:

```
DATE, WS, WD, T, P, H
2023-03-07 00:00, 5.134, 28, 20, 1014, 69.6
2023-03-07 00:10, 5.134, 28, 20, 1014, 69.6
2023-03-07 00:20, 5.134, 28, 20, 1014, 69.6
```

『注意』

1. 脚本运行后会满足条件文件(配置文件中配置的文件名, 且文件修改时间为8秒之前)移动到脚本目录下的tmp/x/src_bak/ (其中x代指配置文件的组号: 第0组则为0, 第1组为1...)
2. 此脚本生成的结果文件在脚本目录下的tmp/x/dst_bak/有备份; 脚本将生成结果文件拷贝到目标目录的过程分两步: (假定文件名为aa.csv)
 - (1) 先将tmp/x/dst_bak/aa.csv拷贝成tmp/x/dst_bak/aa.csv.tmp_copy
 - (2) 将tmp/x/dst_bak/aa.csv.tmp_copy文件移动到目标目录的aa.csv
3. 脚本目录下的tmp目录原则上都只是临时文件的存储, 当配置文件配置了过期删除的设置时, 脚本会把此目录下的所有的过期文件进行清除.
4. 如果程序正常运行, 则程序是长驻内存的; 如果修改配置文件, 程序会自动重新加载配置文件不需要重新启动程序.

软件部署

1.确定要生成的文件所属操作系统用户

比如要生成的文件是在root用户的软件在使用则此文件操作系统用户属主为root

2. 用第1步确定的操作系统用户登录服务器

3. 将merge_windtower_htfile压缩包导入服务器

4. 确定merge_windtower_htfile放于系统哪个目录

比如常用的软件部署目录是 /fglyc/wpfs17/ 则将merge_windtower_htfile压缩包解压后 的文件夹merge_windtower_htfile放于 /fglyc/wpfs17/ 目录下

5. 在部署的目录merge_windtower_htfile下打开配置文件进行配置

例如:你将 merge_windtower_htfile文件夹放于了 /fglyc/wpfs17
则打开 /fglyc/wpfs17/merge_windtower_htfile/cfg/cfg.cfg 文件进行编辑,
此文件有如下几个部分需要配置:

(1) 过期删除设置

```
#####  
#  
#           『 ███ 临时文件及日志过期删除设置 ███ 』  
#  
# 所有的删除设置规则为:(单位为天)  
#    0 不对超期的文件进行删除  
#    >0 表示超过其值的天数日志将要删除  
#  
# g_log_delExpirDays    日志超期删除设置  
# g_tmp_delExpirDays    当前脚本下临时目录tmp下临时文件删除设置  
#  
#####  
g_log_delExpirDays=10  
g_tmp_delExpirDays=1
```

(2) 日志输出控制

```
#####
#
#           『 日志输出控制 』
#
# 调试时      OUT_LOG_LEVEL=${DEBUG}
# 正常运行时 OUT_LOG_LEVEL=${INFO}
#
#
#####
OUT_LOG_LEVEL=${INFO}
```

(3) 结果文件字段长度设置

```
#####
#
#           『 结果文件部分字段长度配置 』
#
# 结果文件形如:cft_001_010_20230307-20230307.csv
# 文件名构成:前缀_测风塔编号_层高_开始日期-结束日期 后缀
#
# 当定义的长度超过实际数值的长度，便会在原值前加0来满足要求
#
# g_fname_wno_len      #定义如上文件中测风塔编号长度
# g_fname_ht_len       #定义如上文件中层高的长度
#
#
#####
g_fname_wno_len="3"
g_fname_ht_len="3"
```

(4) 数据源与结果文件配置

可以配置多组,但至少有一组设置. 要注意的是：多组下标序号必须连续且从0开始.

```
#####
#
#           『 处理文件与结果文件配置 』
#
# g_src_dir          #源文件所在目录
# g_src_file         #源文件名(支持通配符*)
# g_src_nodePrefix   #源文件内容定义一个测风塔编号内容节点的前缀如
#                   某个编号的内容处于<MastData::0001>与
#                   </MastData::0001>之间的内容则此值配置成"MastData::"
# g_src_cntAttrItePrefix #需要从源文件抽取的属性项配置(不带层高),
#                   此处的顺序需要根据结果文件的列顺序来定,
#                   结果文件的内容在此基础上加上时间列即变成
#                   结果文件的一行内容,如可以配置成
#                   "WS,WD,T,P,H"
# g_dst_dir          #结果文件存入目录
#
#   结果文件形如:cft_001_010_20230307-20230307.csv
#   文件名构成:前缀_测风塔编号_层高_开始日期-结束日期 后缀
#
# g_dst_file_prefix  #结果文件前缀如:"cfg"
# g_dst_file_suffix  #结果文件后缀如:".csv"
# g_dst_file_head    #结果文件头,如:"DATE,WS,WD,T,P,H"
# g_dst_time_resolution #结果文件内容时间分辨率(单位:分钟),表示两条数据之间时间间隔
#
#
#####

g_src_dir[0]="/home/fusky/tmp/风机能效/merge_windtower_htfile/gen_tmp_srcfile/result"
g_src_file[0]="JSDLFD_*_CFT.WPD"
g_src_nodePrefix[0]="MastData::"
g_src_cntAttrItePrefix[0]="WS,WD,T,P,H"
g_dst_dir[0]="/home/fusky/tmp/风机能效/merge_windtower_htfile/result"
g_dst_file_prefix[0]="cfg"
g_dst_file_suffix[0]=".csv"
g_dst_file_head[0]="DATE,WS,WD,T,P,H"
g_dst_time_resolution[0]="10"
```

6. 将mergeWtHtFile.sh脚本加入自启动中

将mergeWtHtFile.sh加入自启动中有两种方法:

方法1: 将脚本mergeWtHtFile.sh加入系统的crontab中;

方法2: 将脚本mergeWtHtFile.sh加入自启动软件中,但前提是系统中已经部署了自启动软件.

推荐使用第2种方法

假定系统的自启动软件部署在 /fglyc/wpfs20/startup 下,则打开自启动的的配置文件(配置root用户下的软件自启动
打开 rcfgRoot.cfg文件,其他用户下的软件自启动打开 rcfg.cfg文件) 添加如下配置项:

```
[mergeWtHtFile.sh]
logDir=/fglyc/wpfs17/merge_windtower_htfile/ttylog
runPath=/fglyc/wpfs17/merge_windtower_htfile
runPrePara=
runPara=
```

7. 检查mergeWtHtFile.sh的日志是否有报错及其他情况

- 需要通过命令 `pidof -x mergeWtHtFile.sh` 查看是否有进程在运行(有进程号输出则表示有进程在运行)
- (1) 如果没有进程号则需要排查自启动是否生效，或者脚本重启后自己退出了？
 - (2) 如果程序自己异常退出则需要通过查看日志文件错误信息排除错误

当确认程序进程在运行中后,查看日志文件(当天的日志)是否有ERROR

- (1) 有ERROR 则需要提示信息修改相应部署或配置，直至无ERRR生成
- (2) 没有ERROR时检查结果目录是否有文件生成,源目录是否有源文件生成,生成的结果文件 格式与文件名是否与预期相符合。
- (3) 一切检查都正常后,确认配置文件的日志级别需要是INFO,确认过期文件删除设置是否合理。

软件目录结构说明

软件所在目录的结构和相关说明如下：

```

merge_windtower_htfile/
├─ cfg/
│   └─ cfg.cfg #配置文件
├─ gen_tmp_srcfile/ #脚本自测时用于生成自测数据的目录
│   ├── gen.sh*
│   ├── mode/
│   │   └─ JSDLFD_20110614_1930_CFT.WPD
│   └─ result/
├─ log/ #日志目录
│   ├── mergeWtHtFile_20230412.log
│   ├── mergeWtHtFile_20230413.log
│   └─ mergeWtHtFile_20230414.log
├─ mergeWtHtFile.sh #主功能脚本
├─ myDiyShFunction.sh #自定义脚本函数
├─ readme.md #markdown格式说明文档
├─ result/ #自测脚本时用到的目标目录
│   ├── cfg_001_010_20230412-20230412.csv
│   ├── cfg_001_030_20230412-20230412.csv
│   ├── cfg_001_050_20230412-20230412.csv
│   └─ cfg_002_100_20230412-20230412.csv
├─ sample_output_file/
│   ├── cft_001_010_20230307-20230307.csv #开发此脚本时用到的样例(数据源)
│   └─ JSDLFD_20110614_1930_CFT.WPD #开发此脚本时用到的样例(结果文件)
├─ tmp/ #临时目录
│   └─ 0/ #与配置文件中的组号对应
│       ├── 0001/ #测风塔编号
│       │   ├── col_1.txt #第一列属性对应所有层高数据
│       │   └─ content.txt #存储某个编号对应的源数据
│       ├── 0002/
│       │   ├── col_1.txt
│       │   └─ content.txt
│       ├── dst_bak/ #生成的结果文件备份目录
│       │   ├── cfg_001_010_20230412-20230412.csv
│       │   ├── cfg_001_030_20230412-20230412.csv
│       │   ├── cfg_001_050_20230412-20230412.csv
│       │   ├── cfg_001_060_20230412-20230412.csv
│       │   └─ cfg_002_100_20230412-20230412.csv
│       ├── src_bak/ #要处理的源数据备份目录
│       │   ├── JSDLFD_20230412_1025_CFT.WPD
│       │   ├── JSDLFD_20230412_1030_CFT.WPD
│       │   └─ JSDLFD_20230412_1220_CFT.WPD
│       ├── tmp_cft_no_file.txt #存当前文件的所有测风塔编号
│       └─ tmp_cur_dofiles.txt #源目录下满足条件的文件名
└─ version.txt #当前脚本对应的版本号

```

[返回目录](#)

附配置文件样例

```
#####  
#  
#           『 配置说明 』  
#  
# 1. 此配置文件中凡 '#' 号在一行中开头的行为注释行，不影响配置项  
# 2. 所有配置项所用”号为英文半角状态下输入的引号  
# 3. 配置项中的=号两边不能有空格  
# 4. 所有的路径配置值不要在最后加/符号：  
#     例如：配置/zfmd/wpfs20 此为某个路径值  
#     不能配置成/zfmd/wpfs20/  
#  
#####  
  
#脚本的版本号：  
# 2023-04-11 initial version v20.01.000  
#  
g_version_no="v20.01.000"  
  
  
  
  
#####  
#  
#           『 临时文件及日志过期删除设置 』  
#  
# 所有的删除设置规则为：(单位为天)  
# 0 不对超期的文件进行删除  
# >0 表示超过其值的天数日志将要删除  
#  
# g_log_delExpirDays 日志超期删除设置  
# g_tmp_delExpirDays 当前脚本下临时目录tmp下临时文件删除设置  
#  
#####  
g_log_delExpirDays=10  
g_tmp_delExpirDays=1  
  
  
  
  
#####  
#  
#           『 日志输出控制 』  
#  
# 调试时 OUT_LOG_LEVEL=${DEBUG}
```



```
# 正常运行时 OUT_LOG_LEVEL=${INFO}
#
#
#####

NOOUT=0 ; levelName[0]="NOOUT";
ERROR=1 ; levelName[1]="ERROR";
INFO=2 ; levelName[2]="INFO" ;
DEBUG=3 ; levelName[3]="DEBUG";

#调试时:
#      设置成${DEBUG}
#正常运行时:
#      设置成${INFO}
#
OUT_LOG_LEVEL=${INFO}
#OUT_LOG_LEVEL=${DEBUG}


#####
#
#      『 结果文件部分字段长度配置  』
#
#      结果文件形如:cft_001_010_20230307-20230307.csv
#      文件名构成:前缀_测风塔编号_层高_开始日期-结束日期 后缀
#
#      当定义的长度超过实际数值的长度,便会在原值前加0来满足要求
#
#      g_fname_wno_len      #定义如上文件中测风塔编号长度
#      g_fname_ht_len      #定义如上文件中层高的长度
#
#
#####
g_fname_wno_len="3"
g_fname_ht_len="3"


#####
#
#      『 处理文件与结果文件配置  』
#
#
#      g_src_dir            #源文件所在目录
#      g_src_file          #源文件名(支持通配符*)
#      g_src_nodePrefix    #源文件内容定义一个测风塔编号内容节点的前缀如
#                          某个编号的内容处于<MastData::0001>与
#                          </MastData::0001>之间的内容则此值配置成"MastData::"
#      g_src_cntAttrItePrefix #需要从源文件抽取的属性项配置(不带层高),
```

```

#           此处的顺序需要根据结果文件的列顺序来定，
#           结果文件的内容在此基础上加上时间列即变成
#           结果文件的一行内容,如可以配置成
#           "WS,WD,T,P,H"
# g_dst_dir      #结果文件存入目录
#
#   结果文件形如:cft_001_010_20230307-20230307.csv
#   文件名构成:前缀_测风塔编号_层高_开始日期-结束日期 后缀
#
# g_dst_file_prefix #结果文件前缀如:"cfg"
# g_dst_file_suffix #结果文件后缀如:".csv"
# g_dst_file_head   #结果文件头,如:"DATE,WS,WD,T,P,H"
# g_dst_time_resolution #结果文件内容时间分辨率(单位:分钟),表示两条数据之间时间间隔
#
#

```

```
#####
```

```

g_src_dir[0]="/home/fusky/tmp/风机能效/merge_windtower_htfile/gen_tmp_srcfile/result"
g_src_file[0]="JSDLFD_*_CFT.WPD"
g_src_nodePrefix[0]="MastData::"
g_src_cntAttrItePrefix[0]="WS,WD,T,P,H"
g_dst_dir[0]="/home/fusky/tmp/风机能效/merge_windtower_htfile/result"
g_dst_file_prefix[0]="cfg"
g_dst_file_suffix[0]=".csv"
g_dst_file_head[0]="DATE,WS,WD,T,P,H"
g_dst_time_resolution[0]="10"

```

```
#####
```

```

#
#           『 █ 全局临时变量 █ 』
#

```

```

#   【此部分的配置不需要修改】
#

```

```
#####
```

```

#

```

```

#   【以下配置不需要修改】

```

```

g_grp_nums="${#g_src_dir[*]}"

```