File Parse & Transform

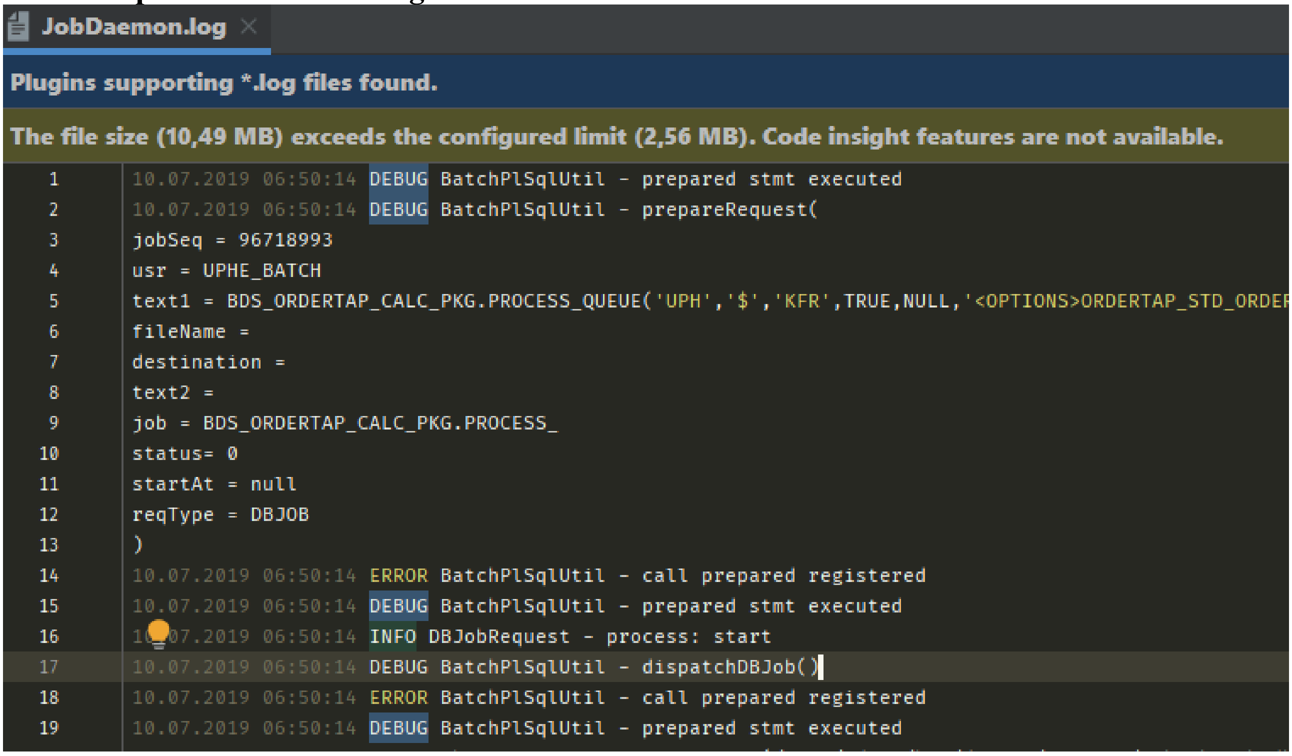
Parse and transform a structured flat file into various formats: xml, xls, csv, txt.
All rules and settings for Parsing & Transformation are defined in json config file.

Parsing:
- Defines certain fields that have to be transformed
- Restricts exported dataset with a list of values
- Validates certain fields against specific datatype (date, time, number)
- Reorders the exported fields
- Rejects unstructured parts of the file

Transformation:
- Defines output file (path, extensions)
- Defines fields titles handling (include or not)
- Defines separator (eg for CSV)
- Defines rowid handling (add or not)

Prerequisite:
- File should have defined structure (it can be some log file or migration structured data file)
- Fields separator should be defined, else blank will be used as a separator

File example: <JobDaemon.log>



Plugins supporting *.log files found.

The file size (10,49 MB) exceeds the configured limit (2,56 MB). Code insight features are not available.

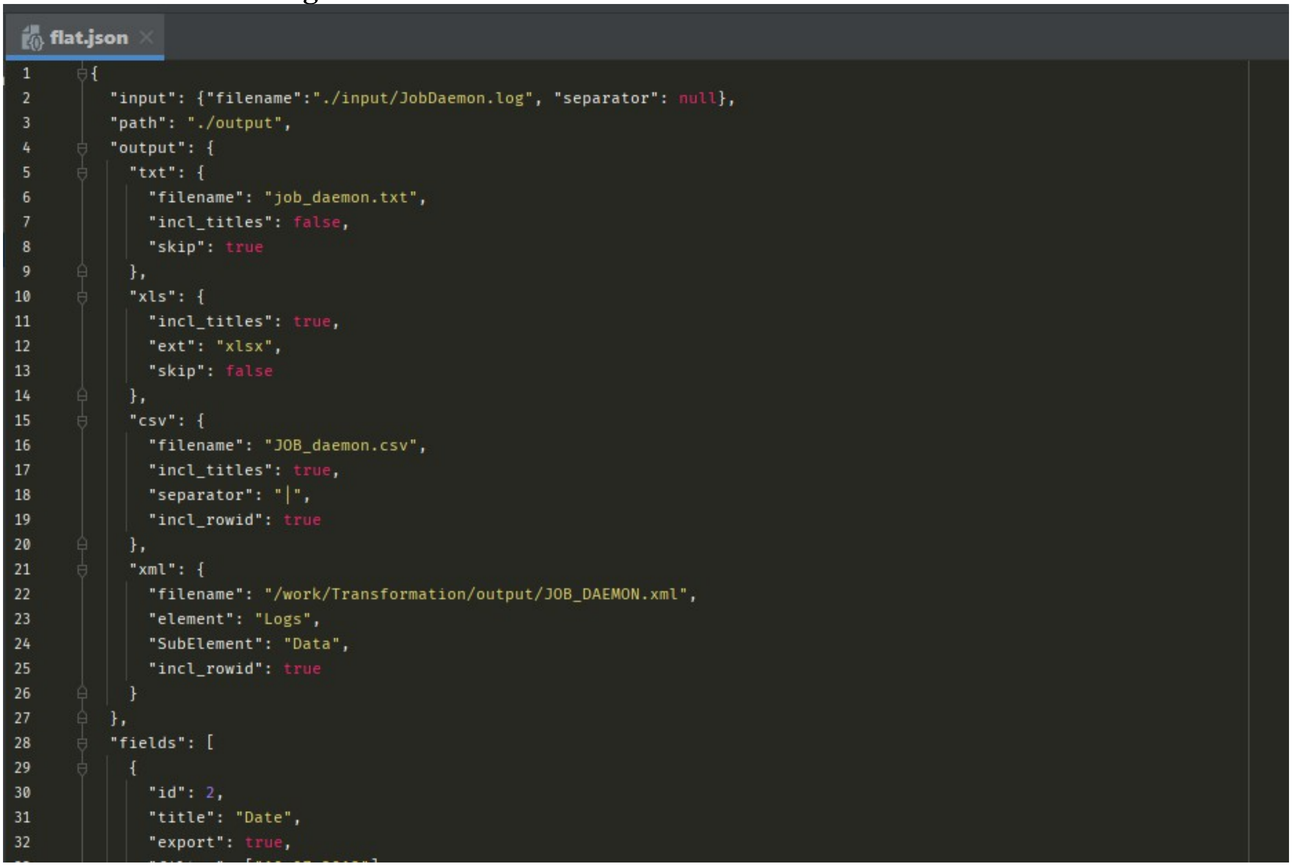
```
10.07.2019 06:50:14 DEBUG BatchPlSqlUtil - prepared stmt executed
10.07.2019 06:50:14 DEBUG BatchPlSqlUtil - prepareRequest(
jobSeq = 96718993
usr = UPHE_BATCH
text1 = BDS_ORDERTAP_CALC_PKG.PROCESS_QUEUE('UPH','$','KFR',TRUE,NULL,'<OPTIONS>ORDERTAP_STD_ORDER
fileName =
destination =
text2 =
job = BDS_ORDERTAP_CALC_PKG.PROCESS_
status= 0
startAt = null
reqType = DBJOB
)
10.07.2019 06:50:14 ERROR BatchPlSqlUtil - call prepared registered
10.07.2019 06:50:14 DEBUG BatchPlSqlUtil - prepared stmt executed
10.07.2019 06:50:14 INFO DBJobRequest - process: start
10.07.2019 06:50:14 DEBUG BatchPlSqlUtil - dispatchDBJob()
10.07.2019 06:50:14 ERROR BatchPlSqlUtil - call prepared registered
10.07.2019 06:50:14 DEBUG BatchPlSqlUtil - prepared stmt executed
```

Sctructure:

4 fields are defined:
–Date (date)
–Time (datetime)
–MesageType (string)
–MessageText (string)
Separator is Blank (null)

Config file <flat.json> (has to be validated - https://codebeautify.org/jsonvalidator)

Transformation settings:



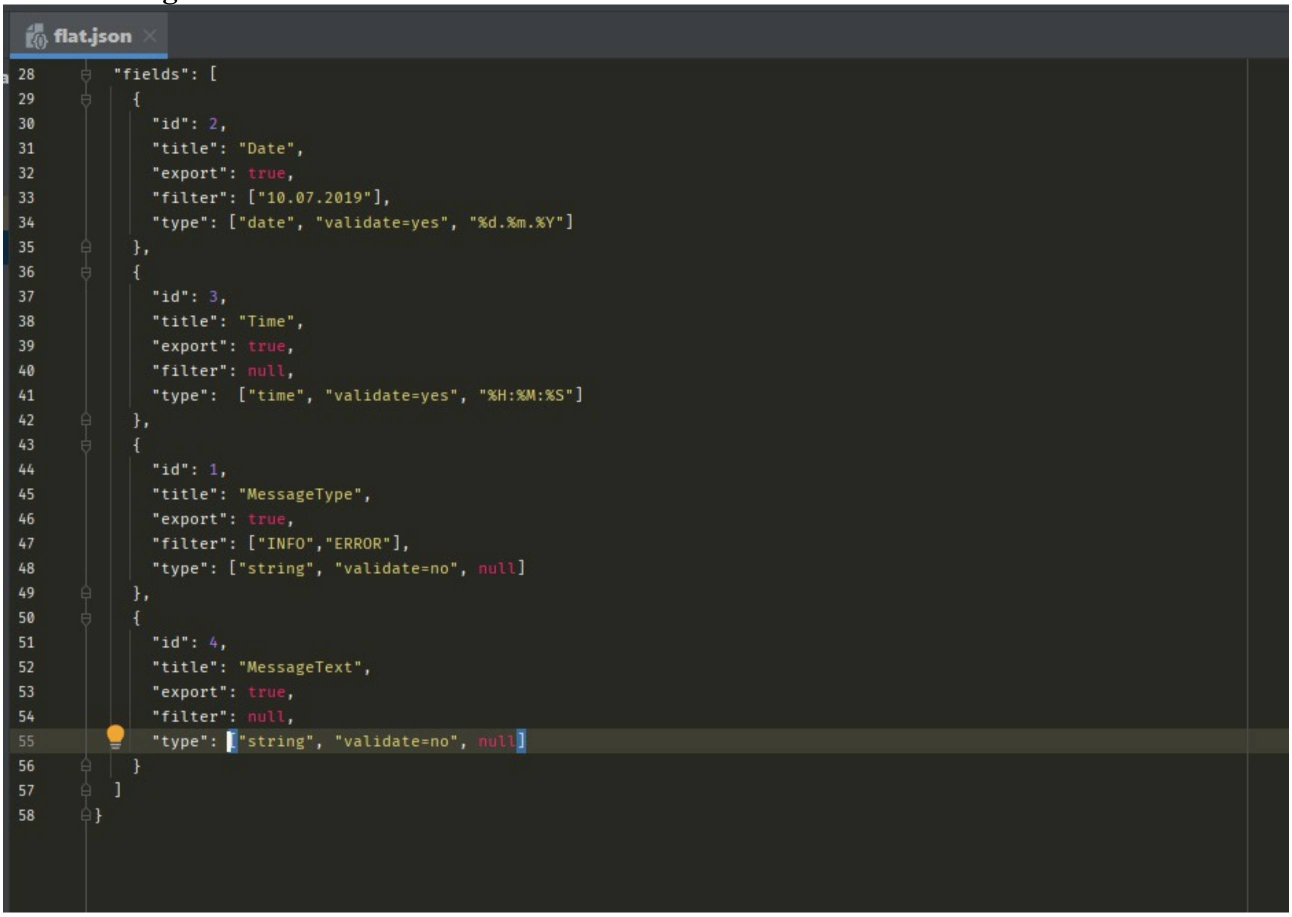
```
{
  "input": {"filename":"./input/JobDaemon.log", "separator": null},
  "path": "./output",
  "output": {
    "txt": {
      "filename": "job_daemon.txt",
      "incl_titles": false,
      "skip": true
    },
    "xls": {
      "incl_titles": true,
      "ext": "xlsx",
      "skip": false
    },
    "csv": {
      "filename": "JOB_daemon.csv",
      "incl_titles": true,
      "separator": "|",
      "incl_rowid": true
    },
    "xml": {
      "filename": "/work/Transformation/output/JOB_DAEMON.xml",
      "element": "Logs",
      "SubElement": "Data",
      "incl_rowid": true
    }
  },
  "fields": [
    {
      "id": 2,
      "title": "Date",
      "export": true,
```

Defines the data export settings (eg: export type, filenames, files elements, some other constraints).

- **input**: hash, that describes input file to be transformed (filename and fields separator, if any)
- **filename**: input filename. If no path provided then path is current dir, ie ./
- **separator**: fields separator of input file, possible values: ("/t" - tab, "|" - pipe, null - blank)
- **path**: directory, where all output files will be stored (used, if not explicitly set per type hash below)
  > it is possible to use UNIX ("/") style (it will be converted to Win style if running from Win)
  > if path starts with root (/work/) this will be automatically substituted with c:\work in Win
- **output**: hash, that defines exported type's setting. Keys in types hashes are not mandatory.
  –**filename**: output filename. If not defined, input filename is used with extension (ext). If file is defined, but without full dir, then global path in previous key (path) will be used
  –**incl_titles**: includes fields titles, defined in fields/title setting (see Parsing Rules below)
    true => export also fields titles in exported file
    false => export data without titles (default)
  –**ext**: file extension. Used only if filename is not provided explicitly. If ext key is not given, then "type" is used as extension (ie for xls: extension will be xls and not xlsx)
  –**skip**: avoid exporting
    true => skip exporting to the given format
    false => do export (default)
  –**separator**: fields separator on output file (utilized only in csv export). Default: blank
  –**element**: xml root element name (default: Log)
  –**SubElement**: xml subelement name (default: Data)

–**incl_rowid**: add extra field “rowid“ (row counter) as the first column in the exported file
true => add rowid fields in the beginning
false => do not add rowid and leave as it is (default)

Parse settings:



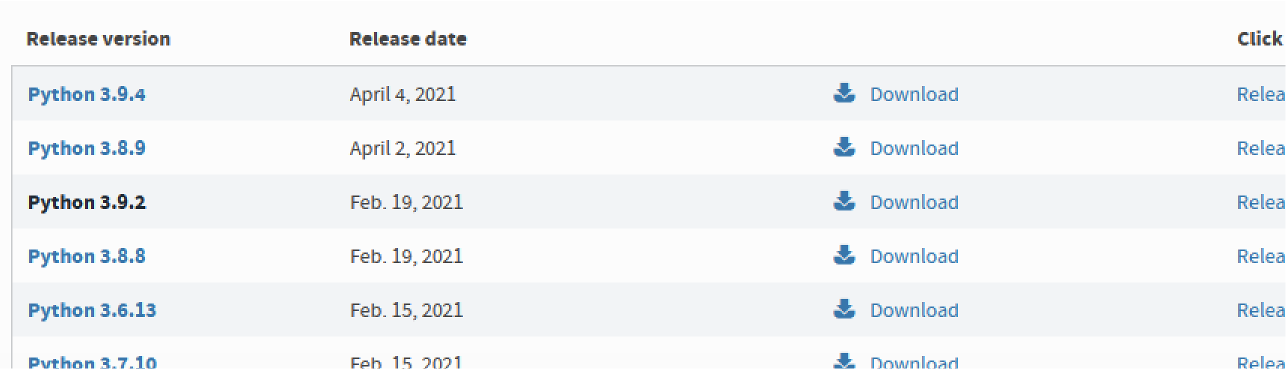
```
  "fields": [
    {
      "id": 2,
      "title": "Date",
      "export": true,
      "filter": ["10.07.2019"],
      "type": ["date", "validate=yes", "%d.%m.%Y"]
    },
    {
      "id": 3,
      "title": "Time",
      "export": true,
      "filter": null,
      "type":  ["time", "validate=yes", "%H:%M:%S"]
    },
    {
      "id": 1,
      "title": "MessageType",
      "export": true,
      "filter": ["INFO","ERROR"],
      "type": ["string", "validate=no", null]
    },
    {
      "id": 4,
      "title": "MessageText",
      "export": true,
      "filter": null,
      "type": ["string", "validate=no", null]
    }
  ]
}
```

Defines the dataset to be transformed, with validation rules and data restriction. All settings are wrapped in the list of fields. The fields in the list are sorted based on defined file structure (see page1). Every field has its out setting/rule as a hash, that handles parsing. All keys are mandatory.

- **id**: sort no that handles field's position in the exported file
- **title**: field's title, that can be included in export (see incl_title key in Transformation rules)
- **export**: decides if the field should be transferred
  true => export the given field
  false => do not export
- **filter**: list of values, restrict the exported data and defines the dataset. From the example above: Export where Date = 10.07.2019 AND MessageType in ('INFO', 'ERROR').
  If export is null then no restriction applied against the given field
- **type**: defines field's type and validation settings. This helps to reject all unstructered part of file (as in the sample input file, seen on the first page).
  validate=yes => validate the given field value against type and format
  validate=no => do not validate the field (all rows will go through)
  From the example above:
  - field Date will be validated against DATE format DD.MM.YYYY
  - field Time will be validate against DATETIME format HH24:SS:MI
  If validation fails, row won't be exported

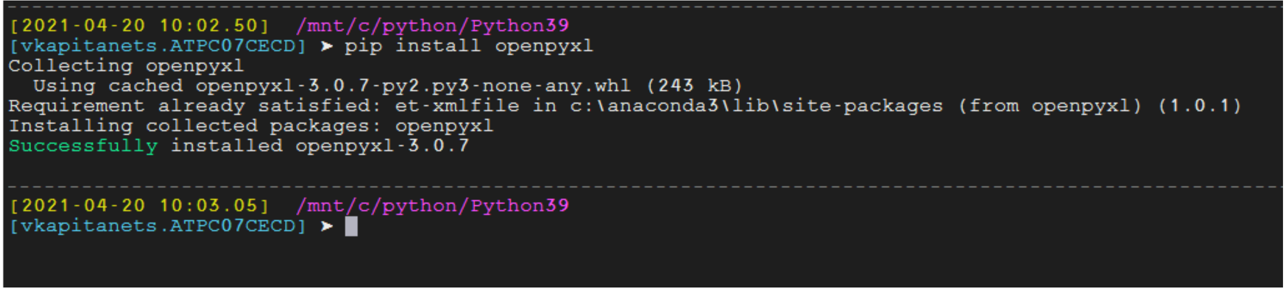
Setup the environment

- install Python 3.9 from https://www.python.org/downloads/



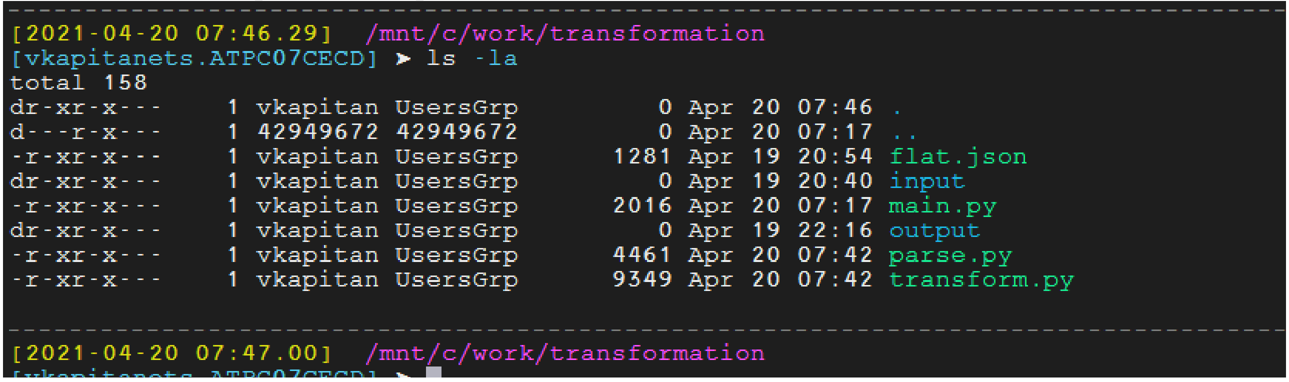
Release version	Release date		Click
Python 3.9.4	April 4, 2021	Download	Relea
Python 3.8.9	April 2, 2021	Download	Relea
Python 3.9.2	Feb. 19, 2021	Download	Relea
Python 3.8.8	Feb. 19, 2021	Download	Relea
Python 3.6.13	Feb. 15, 2021	Download	Relea
Python 3.7.10	Feb. 15, 2021	Download	Relea

- for xls integration, install openpyxl module (all other used modules are part of python installation)
    $ pip install openpyxl



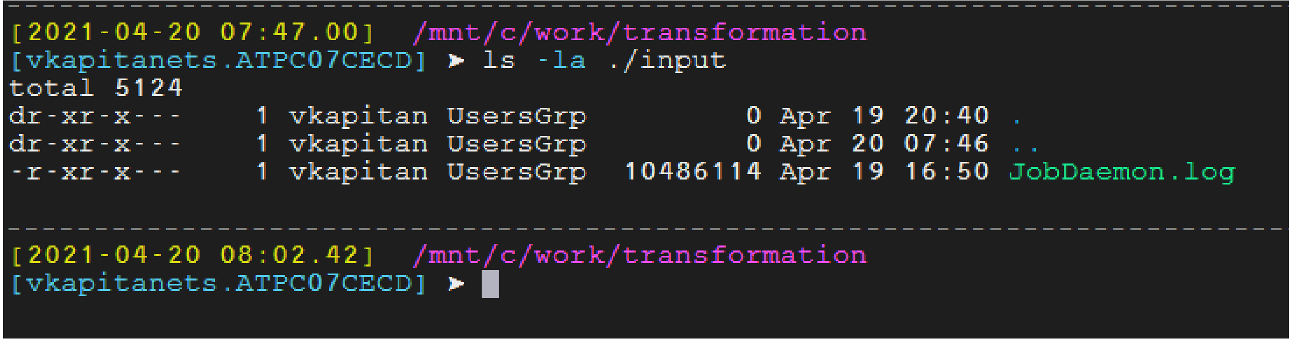
```
[2021-04-20 10:02.50]  /mnt/c/python/Python39
[vkapitanets.ATPC07CECD] ➤ pip install openpyxl
Collecting openpyxl
  Using cached openpyxl-3.0.7-py2.py3-none-any.whl (243 kB)
Requirement already satisfied: et-xmlfile in c:\anaconda3\lib\site-packages (from openpyxl) (1.0.1)
Installing collected packages: openpyxl
Successfully installed openpyxl-3.0.7

[2021-04-20 10:03.05]  /mnt/c/python/Python39
[vkapitanets.ATPC07CECD] ➤
```

- create /work/transformation and put there flat.json, parse.py, transform.py and main.py source files
- create subdirs: ./input and ./output



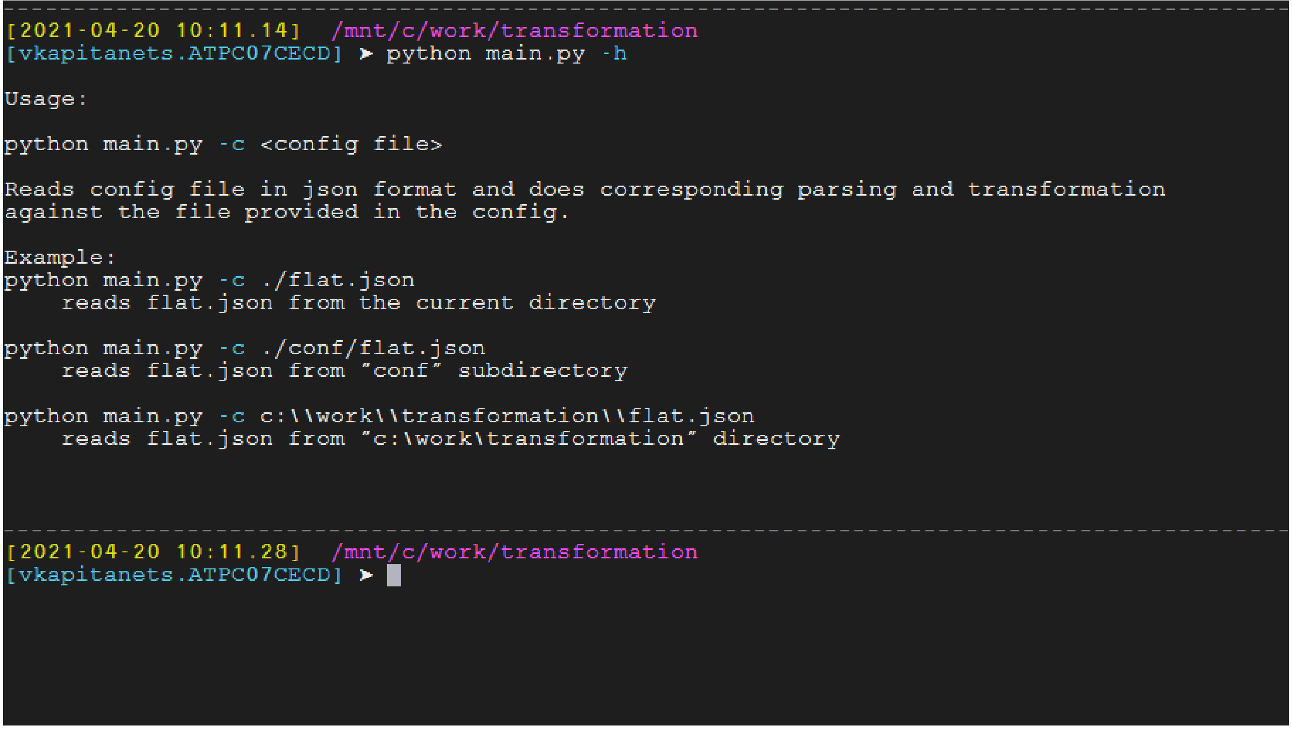
```
[2021-04-20 07:46.29]  /mnt/c/work/transformation
[vkapitanets.ATPC07CECD] ➤ ls -la
total 158
dr-xr-x---    1 vkapitan UsersGrp          0 Apr 20 07:46 .
d---r-x---    1 42949672 42949672          0 Apr 20 07:17 ..
-r-xr-x---    1 vkapitan UsersGrp       1281 Apr 19 20:54 flat.json
dr-xr-x---    1 vkapitan UsersGrp          0 Apr 19 20:40 input
-r-xr-x---    1 vkapitan UsersGrp       2016 Apr 20 07:17 main.py
dr-xr-x---    1 vkapitan UsersGrp          0 Apr 19 22:16 output
-r-xr-x---    1 vkapitan UsersGrp       4461 Apr 20 07:42 parse.py
-r-xr-x---    1 vkapitan UsersGrp       9349 Apr 20 07:42 transform.py

[2021-04-20 07:47.00]  /mnt/c/work/transformation
[vkapitanets.ATPC07CECD] ➤
```

- put JobDaemon.log file into ./input directory

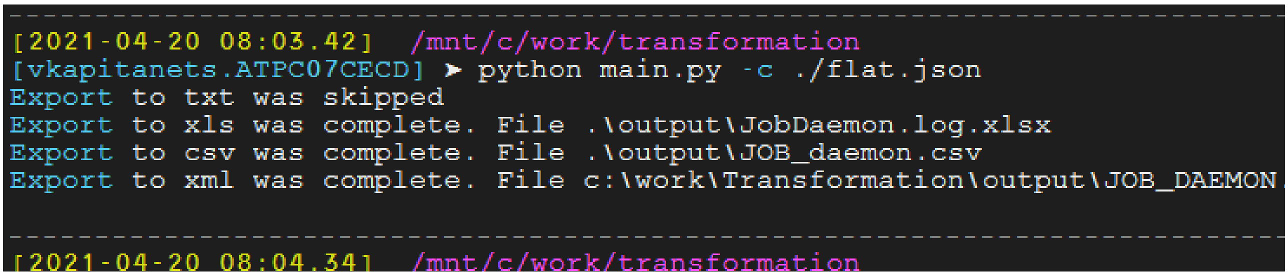


```
[2021-04-20 07:47.00]  /mnt/c/work/transformation
[vkapitanets.ATPC07CECD] ➤ ls -la ./input
total 5124
dr-xr-x---    1 vkapitan UsersGrp          0 Apr 19 20:40 .
dr-xr-x---    1 vkapitan UsersGrp          0 Apr 20 07:46 ..
-r-xr-x---    1 vkapitan UsersGrp   10486114 Apr 19 16:50 JobDaemon.log

[2021-04-20 08:02.42]  /mnt/c/work/transformation
[vkapitanets.ATPC07CECD] ➤
```

- run the script

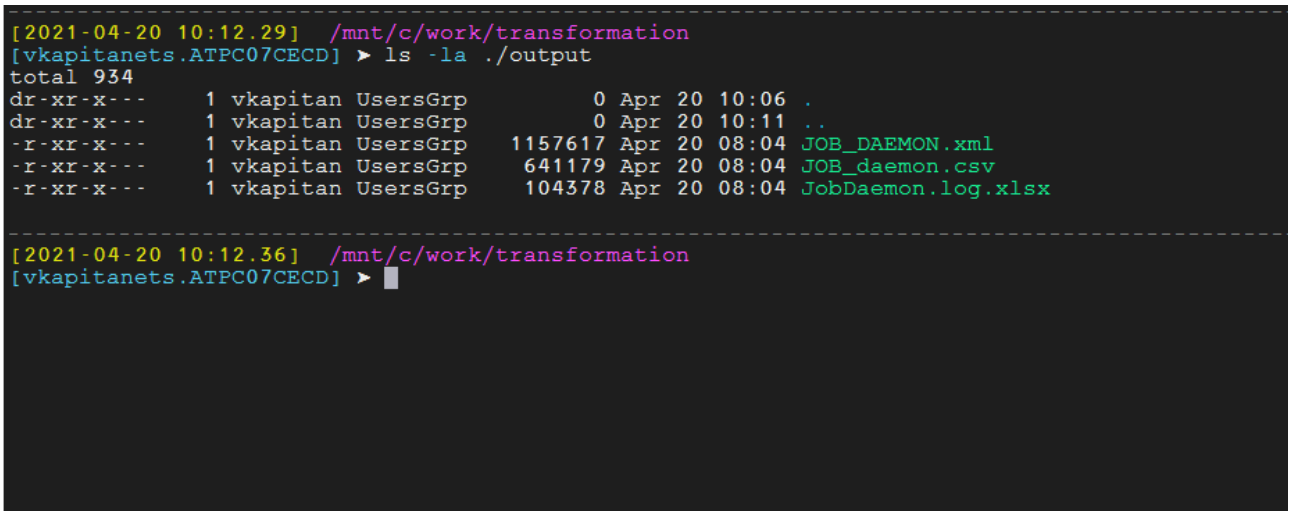


```
-------------------------------------------------------------------------------------------
[2021-04-20 10:11.14]  /mnt/c/work/transformation
[vkapitanets.ATPC07CECD] ➤ python main.py -h

Usage:

python main.py -c <config file>

Reads config file in json format and does corresponding parsing and transformation
against the file provided in the config.

Example:
python main.py -c ./flat.json
    reads flat.json from the current directory

python main.py -c ./conf/flat.json
    reads flat.json from ”conf” subdirectory

python main.py -c c:\\work\\transformation\\flat.json
    reads flat.json from ”c:\work\transformation” directory



-------------------------------------------------------------------------------------------
[2021-04-20 10:11.28]  /mnt/c/work/transformation
[vkapitanets.ATPC07CECD] ➤ █
```



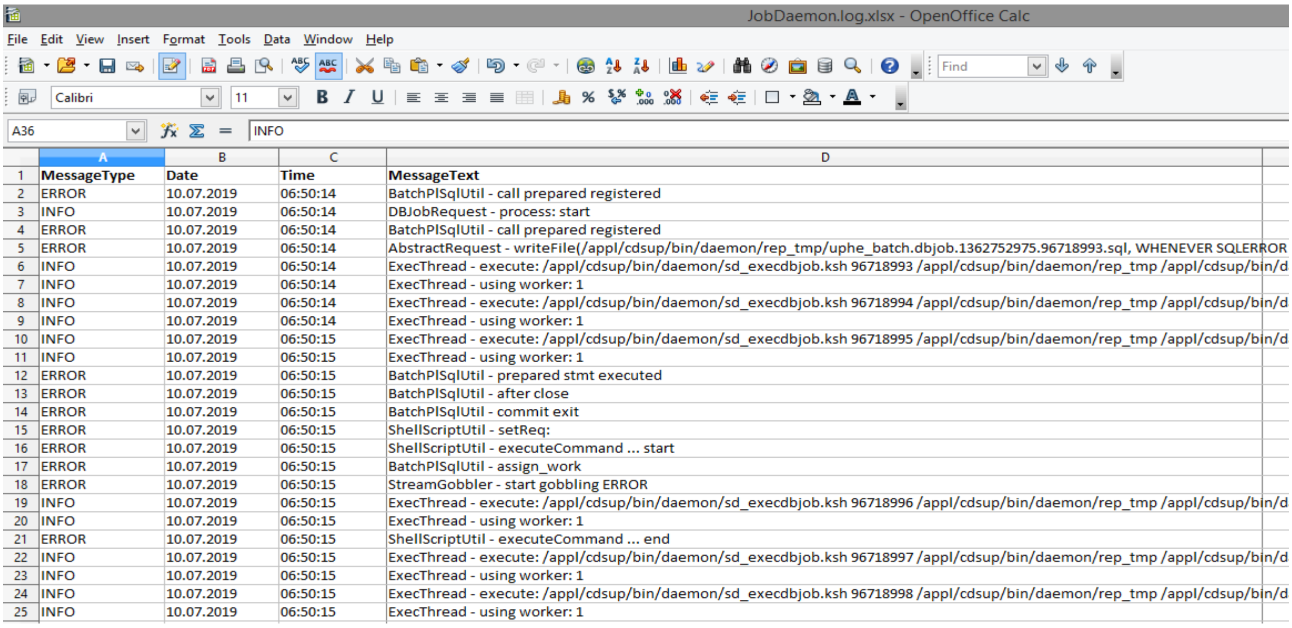
```
-----------------------------------------------------------------------------
[2021-04-20 08:03.42]  /mnt/c/work/transformation
[vkapitanets.ATPC07CECD] ➤ python main.py -c ./flat.json
Export to txt was skipped
Export to xls was complete. File .\output\JobDaemon.log.xlsx
Export to csv was complete. File .\output\JOB_daemon.csv
Export to xml was complete. File c:\work\Transformation\output\JOB_DAEMON.

-----------------------------------------------------------------------------
[2021-04-20 08:04.34]  /mnt/c/work/transformation
```

- check the results

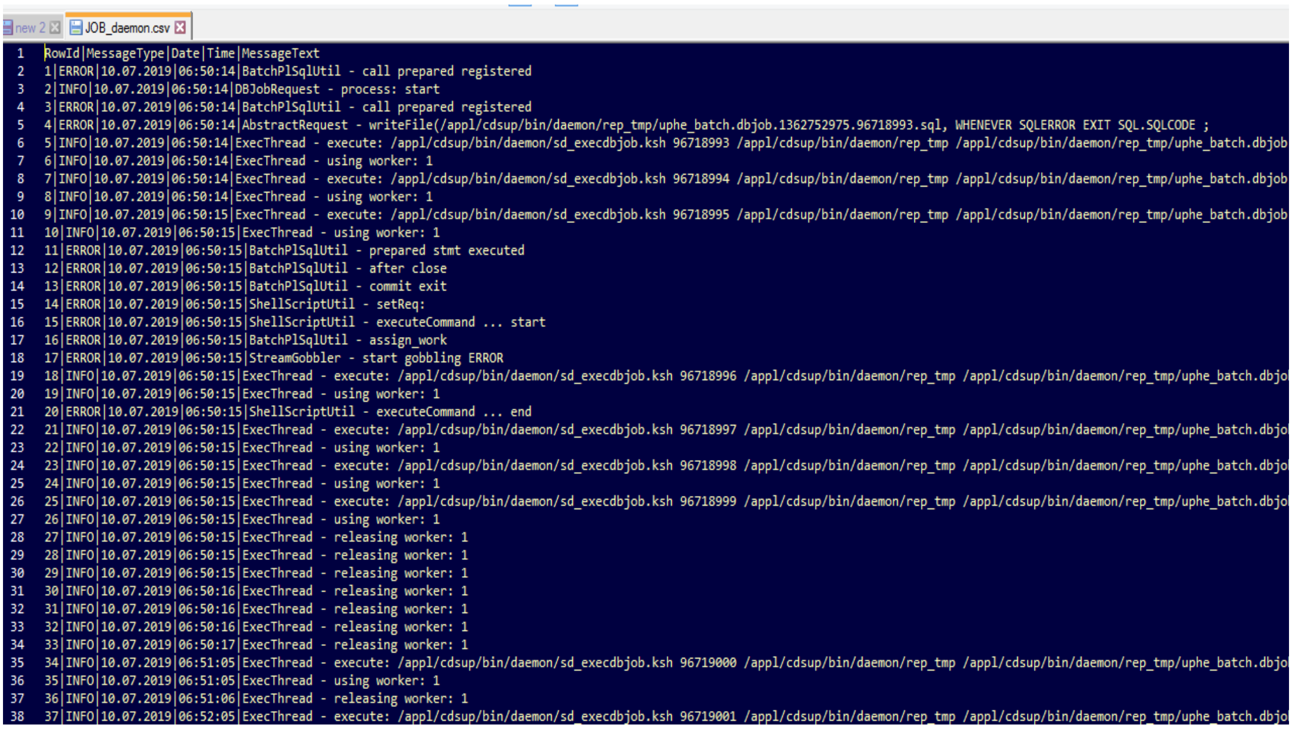


```
-------------------------------------------------------------------------------------------
[2021-04-20 10:12.29]  /mnt/c/work/transformation
[vkapitanets.ATPC07CECD] ➤ ls -la ./output
total 934
dr-xr-x---    1 vkapitan UsersGrp         0 Apr 20 10:06 .
dr-xr-x---    1 vkapitan UsersGrp         0 Apr 20 10:11 ..
-r-xr-x---    1 vkapitan UsersGrp   1157617 Apr 20 08:04 JOB_DAEMON.xml
-r-xr-x---    1 vkapitan UsersGrp    641179 Apr 20 08:04 JOB_daemon.csv
-r-xr-x---    1 vkapitan UsersGrp    104378 Apr 20 08:04 JobDaemon.log.xlsx

-------------------------------------------------------------------------------------------
[2021-04-20 10:12.36]  /mnt/c/work/transformation
[vkapitanets.ATPC07CECD] ➤ █
```

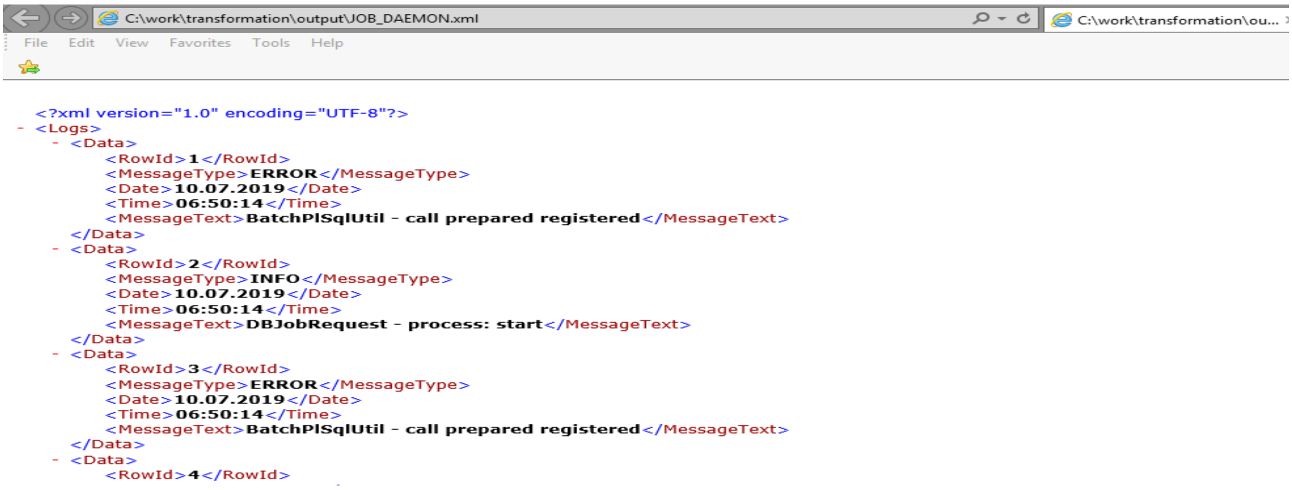
Results:


JobDaemon.log.xlsx - OpenOffice Calc
File Edit View Insert Format Tools Data Window Help
Find
Calibri
11
A36
INFO

	A	B	C	D
1	MessageType	Date	Time	MessageText
2	ERROR	10.07.2019	06:50:14	BatchPlSqlUtil - call prepared registered
3	INFO	10.07.2019	06:50:14	DBJobRequest - process: start
4	ERROR	10.07.2019	06:50:14	BatchPlSqlUtil - call prepared registered
5	ERROR	10.07.2019	06:50:14	AbstractRequest - writeFile(/appl/cdsup/bin/daemon/rep_tmp/uphe_batch.dbjob.1362752975.96718993.sql, WHENEVER SQLERROR
6	INFO	10.07.2019	06:50:14	ExecThread - execute: /appl/cdsup/bin/daemon/sd_execdbjob.ksh 96718993 /appl/cdsup/bin/daemon/rep_tmp /appl/cdsup/bin/d
7	INFO	10.07.2019	06:50:14	ExecThread - using worker: 1
8	INFO	10.07.2019	06:50:14	ExecThread - execute: /appl/cdsup/bin/daemon/sd_execdbjob.ksh 96718994 /appl/cdsup/bin/daemon/rep_tmp /appl/cdsup/bin/d
9	INFO	10.07.2019	06:50:14	ExecThread - using worker: 1
10	INFO	10.07.2019	06:50:15	ExecThread - execute: /appl/cdsup/bin/daemon/sd_execdbjob.ksh 96718995 /appl/cdsup/bin/daemon/rep_tmp /appl/cdsup/bin/d
11	INFO	10.07.2019	06:50:15	ExecThread - using worker: 1
12	ERROR	10.07.2019	06:50:15	BatchPlSqlUtil - prepared stmt executed
13	ERROR	10.07.2019	06:50:15	BatchPlSqlUtil - after close
14	ERROR	10.07.2019	06:50:15	BatchPlSqlUtil - commit exit
15	ERROR	10.07.2019	06:50:15	ShellScriptUtil - setReq:
16	ERROR	10.07.2019	06:50:15	ShellScriptUtil - executeCommand ... start
17	ERROR	10.07.2019	06:50:15	BatchPlSqlUtil - assign_work
18	ERROR	10.07.2019	06:50:15	StreamGobbler - start gobbling ERROR
19	INFO	10.07.2019	06:50:15	ExecThread - execute: /appl/cdsup/bin/daemon/sd_execdbjob.ksh 96718996 /appl/cdsup/bin/daemon/rep_tmp /appl/cdsup/bin/d
20	INFO	10.07.2019	06:50:15	ExecThread - using worker: 1
21	ERROR	10.07.2019	06:50:15	ShellScriptUtil - executeCommand ... end
22	INFO	10.07.2019	06:50:15	ExecThread - execute: /appl/cdsup/bin/daemon/sd_execdbjob.ksh 96718997 /appl/cdsup/bin/daemon/rep_tmp /appl/cdsup/bin/d
23	INFO	10.07.2019	06:50:15	ExecThread - using worker: 1
24	INFO	10.07.2019	06:50:15	ExecThread - execute: /appl/cdsup/bin/daemon/sd_execdbjob.ksh 96718998 /appl/cdsup/bin/daemon/rep_tmp /appl/cdsup/bin/d
25	INFO	10.07.2019	06:50:15	ExecThread - using worker: 1



```
RowId|MessageType|Date|Time|MessageText
1|ERROR|10.07.2019|06:50:14|BatchPlSqlUtil - call prepared registered
2|INFO|10.07.2019|06:50:14|DBJobRequest - process: start
3|ERROR|10.07.2019|06:50:14|BatchPlSqlUtil - call prepared registered
4|ERROR|10.07.2019|06:50:14|AbstractRequest - writeFile(/appl/cdsup/bin/daemon/rep_tmp/uphe_batch.dbjob.1362752975.96718993.sql, WHENEVER SQLERROR EXIT SQL.SQLCODE ;
5|INFO|10.07.2019|06:50:14|ExecThread - execute: /appl/cdsup/bin/daemon/sd_execdbjob.ksh 96718993 /appl/cdsup/bin/daemon/rep_tmp /appl/cdsup/bin/daemon/rep_tmp/uphe_batch.dbjob
6|INFO|10.07.2019|06:50:14|ExecThread - using worker: 1
7|INFO|10.07.2019|06:50:14|ExecThread - execute: /appl/cdsup/bin/daemon/sd_execdbjob.ksh 96718994 /appl/cdsup/bin/daemon/rep_tmp /appl/cdsup/bin/daemon/rep_tmp/uphe_batch.dbjob
8|INFO|10.07.2019|06:50:14|ExecThread - using worker: 1
9|INFO|10.07.2019|06:50:15|ExecThread - execute: /appl/cdsup/bin/daemon/sd_execdbjob.ksh 96718995 /appl/cdsup/bin/daemon/rep_tmp /appl/cdsup/bin/daemon/rep_tmp/uphe_batch.dbjob
10|INFO|10.07.2019|06:50:15|ExecThread - using worker: 1
11|ERROR|10.07.2019|06:50:15|BatchPlSqlUtil - prepared stmt executed
12|ERROR|10.07.2019|06:50:15|BatchPlSqlUtil - after close
13|ERROR|10.07.2019|06:50:15|BatchPlSqlUtil - commit exit
14|ERROR|10.07.2019|06:50:15|ShellScriptUtil - setReq:
15|ERROR|10.07.2019|06:50:15|ShellScriptUtil - executeCommand ... start
16|ERROR|10.07.2019|06:50:15|BatchPlSqlUtil - assign_work
17|ERROR|10.07.2019|06:50:15|StreamGobbler - start gobbling ERROR
18|INFO|10.07.2019|06:50:15|ExecThread - execute: /appl/cdsup/bin/daemon/sd_execdbjob.ksh 96718996 /appl/cdsup/bin/daemon/rep_tmp /appl/cdsup/bin/daemon/rep_tmp/uphe_batch.dbjo
19|INFO|10.07.2019|06:50:15|ExecThread - using worker: 1
20|ERROR|10.07.2019|06:50:15|ShellScriptUtil - executeCommand ... end
21|INFO|10.07.2019|06:50:15|ExecThread - execute: /appl/cdsup/bin/daemon/sd_execdbjob.ksh 96718997 /appl/cdsup/bin/daemon/rep_tmp /appl/cdsup/bin/daemon/rep_tmp/uphe_batch.dbjo
22|INFO|10.07.2019|06:50:15|ExecThread - using worker: 1
23|INFO|10.07.2019|06:50:15|ExecThread - execute: /appl/cdsup/bin/daemon/sd_execdbjob.ksh 96718998 /appl/cdsup/bin/daemon/rep_tmp /appl/cdsup/bin/daemon/rep_tmp/uphe_batch.dbjo
24|INFO|10.07.2019|06:50:15|ExecThread - using worker: 1
25|INFO|10.07.2019|06:50:15|ExecThread - execute: /appl/cdsup/bin/daemon/sd_execdbjob.ksh 96718999 /appl/cdsup/bin/daemon/rep_tmp /appl/cdsup/bin/daemon/rep_tmp/uphe_batch.dbjo
26|INFO|10.07.2019|06:50:15|ExecThread - using worker: 1
27|INFO|10.07.2019|06:50:15|ExecThread - releasing worker: 1
28|INFO|10.07.2019|06:50:15|ExecThread - releasing worker: 1
29|INFO|10.07.2019|06:50:15|ExecThread - releasing worker: 1
30|INFO|10.07.2019|06:50:16|ExecThread - releasing worker: 1
31|INFO|10.07.2019|06:50:16|ExecThread - releasing worker: 1
32|INFO|10.07.2019|06:50:16|ExecThread - releasing worker: 1
33|INFO|10.07.2019|06:50:17|ExecThread - releasing worker: 1
34|INFO|10.07.2019|06:51:05|ExecThread - execute: /appl/cdsup/bin/daemon/sd_execdbjob.ksh 96719000 /appl/cdsup/bin/daemon/rep_tmp /appl/cdsup/bin/daemon/rep_tmp/uphe_batch.dbjo
35|INFO|10.07.2019|06:51:05|ExecThread - using worker: 1
36|INFO|10.07.2019|06:51:06|ExecThread - releasing worker: 1
37|INFO|10.07.2019|06:52:05|ExecThread - execute: /appl/cdsup/bin/daemon/sd_execdbjob.ksh 96719001 /appl/cdsup/bin/daemon/rep_tmp /appl/cdsup/bin/daemon/rep_tmp/uphe_batch.dbjo
```


C:\work\transformation\output\JOB_DAEMON.xml
C:\work\transformation\ou...
File Edit View Favorites Tools Help

```
  <?xml version="1.0" encoding="UTF-8"?>
- <Logs>
  - <Data>
        <RowId>1</RowId>
        <MessageType>ERROR</MessageType>
        <Date>10.07.2019</Date>
        <Time>06:50:14</Time>
        <MessageText>BatchPlSqlUtil - call prepared registered</MessageText>
    </Data>
  - <Data>
        <RowId>2</RowId>
        <MessageType>INFO</MessageType>
        <Date>10.07.2019</Date>
        <Time>06:50:14</Time>
        <MessageText>DBJobRequest - process: start</MessageText>
    </Data>
  - <Data>
        <RowId>3</RowId>
        <MessageType>ERROR</MessageType>
        <Date>10.07.2019</Date>
        <Time>06:50:14</Time>
        <MessageText>BatchPlSqlUtil - call prepared registered</MessageText>
    </Data>
  - <Data>
        <RowId>4</RowId>
```