

Check Ottoview logging file and find Errors Procedure

Purpose: This procedure outlines the steps about how to process logging files when you got logs from Linux/Ottoview. It will find all if Errors in all of logging files which are ruby files, sensor recording database, zip files and so on. It will help us to know which sensor and how many sensors have issues when Ottoview doesn't run or running is not good.

Operating Systems: Ubuntu Linux/ VM Ware

Setup Location: Linux Desktop

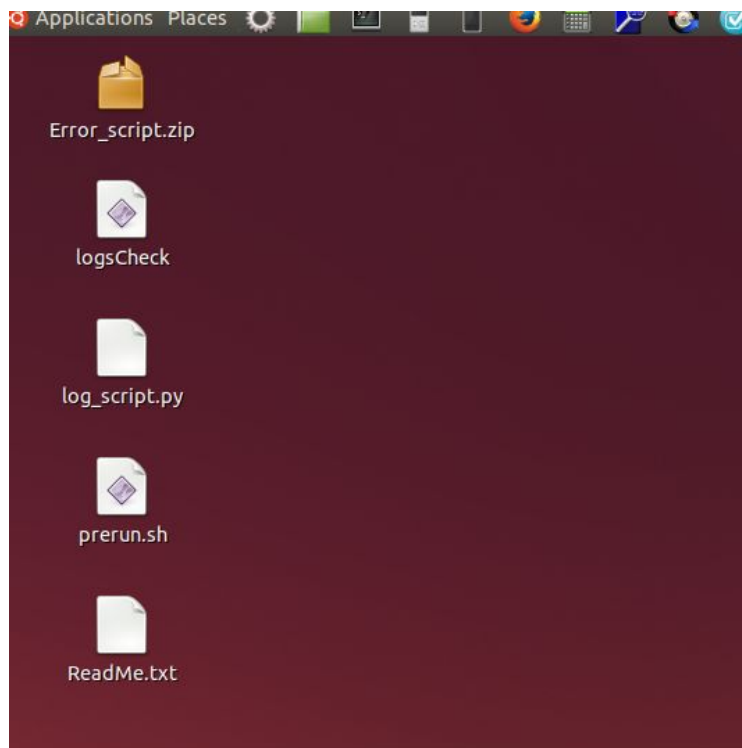
Required Software: Python3, Un-zip tools

Required Files: Error_Script.zip (Confluence link: [Confluence link](http://confluenceprod1.delphiauto.net:8090/display/ADHWP/Ottoview+logging+Errors+checking+script) or <http://confluenceprod1.delphiauto.net:8090/display/ADHWP/Ottoview+logging+Errors+checking+script>), Vehicle - Ottoview logging file

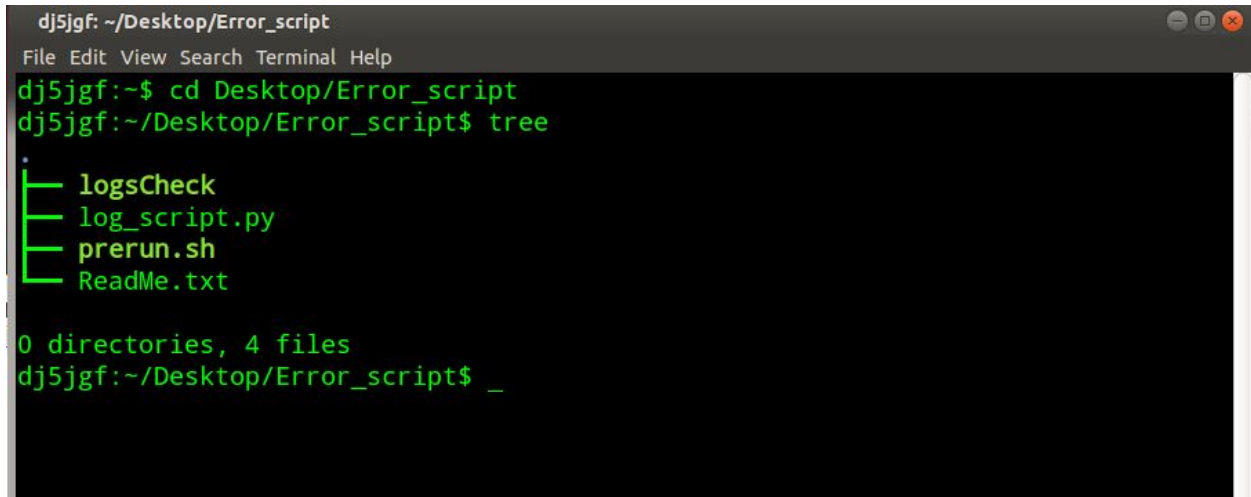
Required Hardware: Working PC Installed with VMWare & Ubuntu, & have properly prepared per "Guide to get a computer ready.pdf" from PTC.

Running procedure

1. Un-zip file Error_Script.zip and copy all of these files to your Linux Desktop

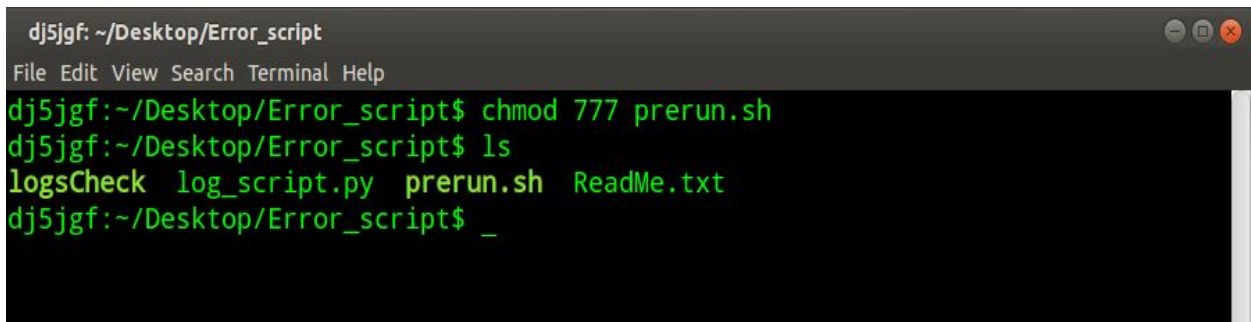


2. Open Terminal:Type the following: Ctrl+Alt+'T'(Keyboard), in the Terminal:Type the following:
cd Desktop/Error_script

A terminal window titled 'dj5jgf: ~/Desktop/Error_script' with a menu bar (File, Edit, View, Search, Terminal, Help). The prompt is 'dj5jgf:~\$'. The user enters 'cd Desktop/Error_script' and then 'tree'. The output shows a directory tree with files: logsCheck, log_script.py, prerun.sh, and ReadMe.txt. Below the tree, it says '0 directories, 4 files'. The prompt returns to 'dj5jgf:~/Desktop/Error_script\$ _'.

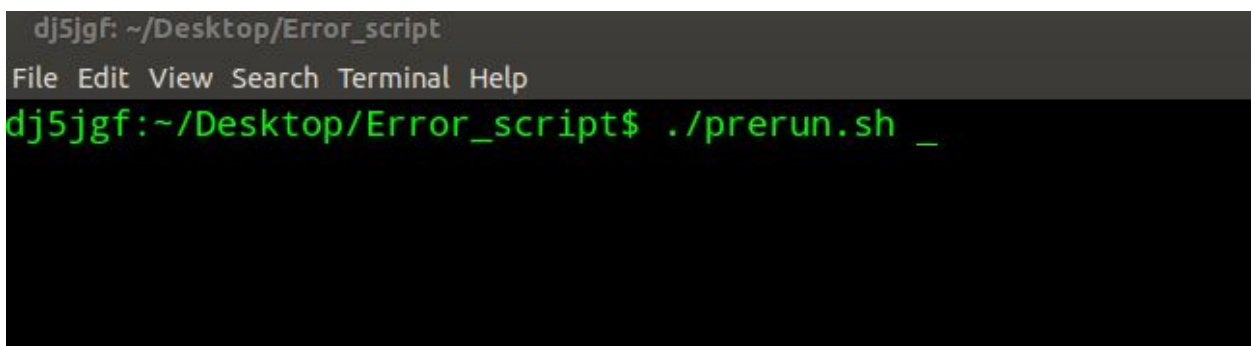
You will see you have files like figure above.

3. In the terminal, type the following: chmod 777 prerun.sh

A terminal window titled 'dj5jgf: ~/Desktop/Error_script' with a menu bar (File, Edit, View, Search, Terminal, Help). The prompt is 'dj5jgf:~/Desktop/Error_script\$'. The user enters 'chmod 777 prerun.sh'. The prompt returns to 'dj5jgf:~/Desktop/Error_script\$'. The user then enters 'ls', and the output shows 'logsCheck log_script.py prerun.sh ReadMe.txt'. The prompt returns to 'dj5jgf:~/Desktop/Error_script\$ _'.

This will help you get access to files with current user logging in credentials

4. In the terminal, type the following: ./prerun.sh

A terminal window titled 'dj5jgf: ~/Desktop/Error_script' with a menu bar (File, Edit, View, Search, Terminal, Help). The prompt is 'dj5jgf:~/Desktop/Error_script\$'. The user enters './prerun.sh'. The prompt returns to 'dj5jgf:~/Desktop/Error_script\$ _'.

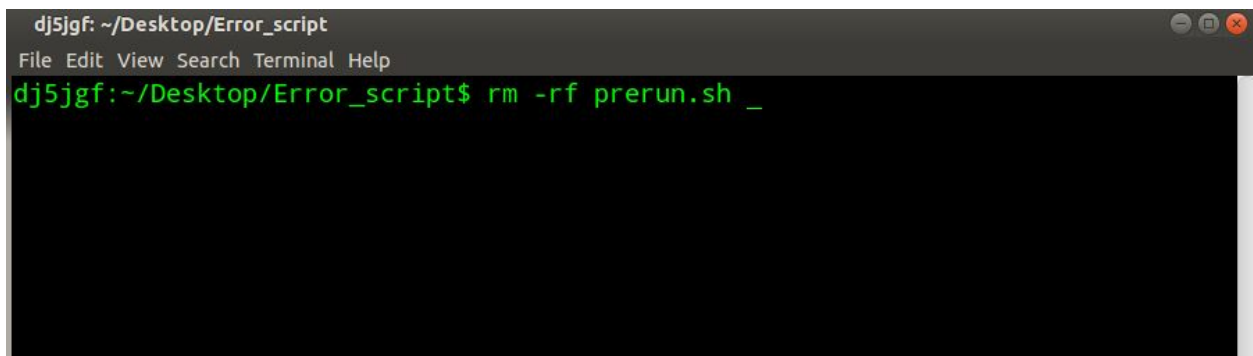
This line will run the following (2) commands:

```
$ mv logsCheck ~/wkspaces/90210123_Otto2ECTrans_Package/scripts/
```

```
$ chmod 777 ~/wkspaces/90210123_Otto2ECTrans_Package/scripts/logsCheck
```

It will copy Linux bash code to the Ottoview scripts path, which allocated in .bashrc, and also get current user access for bash scripts.

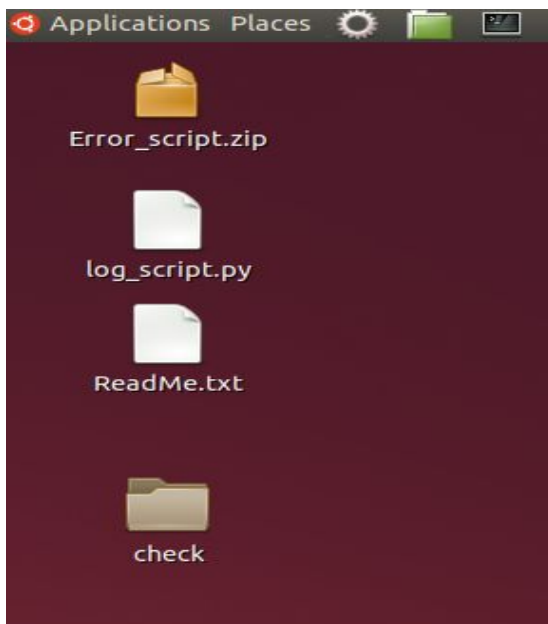
5. In the terminal, type the following:: `rm -rf prerun.sh`



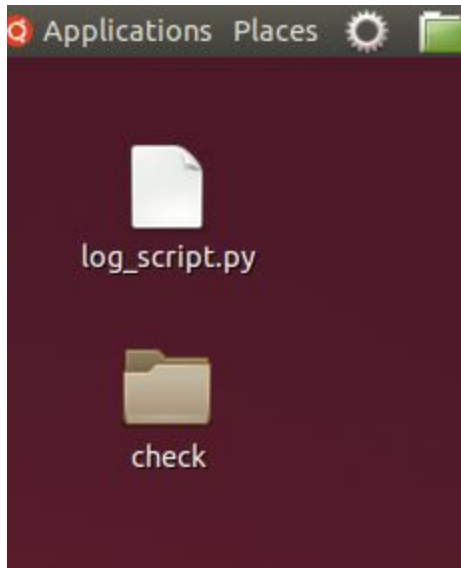
It will permanently delete prerun.sh file. Because you do not need it in the future, Note: -rf is very powerful use with extreme caching.

6. Close Terminal, type the following:: Alt + F4, or using mouse click x

7.
 - a) Copy your logging folder to the Desktop
 - b) Change your logging folder name (e.g. 2018.07.20.09.48.30_pac-veh-ten) to 'check' so, right now your folder name is 'check'



8. Until now, you are basically done, you can delete files that are not needed files like zip files and readme.txt and so on. Just keep the folder named “check” and log_script.py.



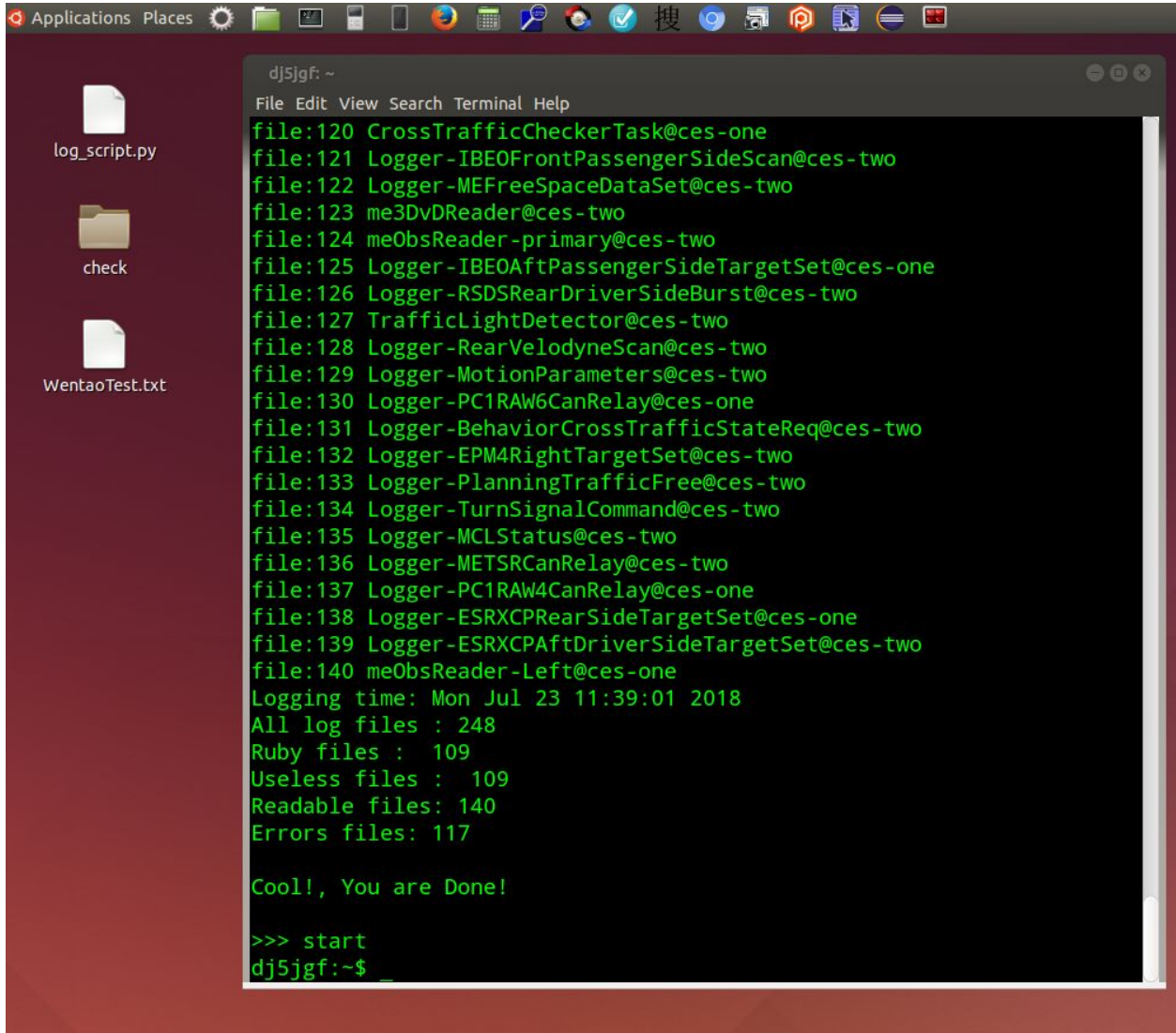
9. open Terminal, type the following:: logsCheck, read instruction in the terminal

```
dj5jgf: ~  
File Edit View Search Terminal Help  
dj5jgf:~$ logsCheck  
Publisher: wen, version 1.0  
  
Usage: logsCheck [Options]  
  -wen or -w: extract errors in logging  
  -help or -h: more details of scripts  
  
Working Example:  
  Terminal: logsCheck -wen  
  ... will process Error loggings and send it to Desktop  
  
dj5jgf:~$ _
```

10. In the Terminal, type the following: logsCheck -wen

a) As you can see from below, It check each files from folder that you set, and look for Error and write it to the new txt file.

b) You will see ErrorTest.txt which contains Error



```
dj5jgf: ~
File Edit View Search Terminal Help
file:120 CrossTrafficCheckerTask@ces-one
file:121 Logger-IBEOFrontPassengerSideScan@ces-two
file:122 Logger-MEFreeSpaceDataSet@ces-two
file:123 me3DvDReader@ces-two
file:124 meObsReader-primary@ces-two
file:125 Logger-IBEOAftPassengerSideTargetSet@ces-one
file:126 Logger-RSDSRearDriverSideBurst@ces-two
file:127 TrafficLightDetector@ces-two
file:128 Logger-RearVelodyneScan@ces-two
file:129 Logger-MotionParameters@ces-two
file:130 Logger-PC1RAW6CanRelay@ces-one
file:131 Logger-BehaviorCrossTrafficStateReq@ces-two
file:132 Logger-EP4RightTargetSet@ces-two
file:133 Logger-PlanningTrafficFree@ces-two
file:134 Logger-TurnSignalCommand@ces-two
file:135 Logger-MCLStatus@ces-two
file:136 Logger-METSRCanRelay@ces-two
file:137 Logger-PC1RAW4CanRelay@ces-one
file:138 Logger-ESRXCP RearSideTargetSet@ces-one
file:139 Logger-ESRXCPAftDriverSideTargetSet@ces-two
file:140 meObsReader-Left@ces-one
Logging time: Mon Jul 23 11:39:01 2018
All log files : 248
Ruby files : 109
Useless files : 109
Readable files: 140
Errors files: 117

Cool!, You are Done!

>>> start
dj5jgf:~$
```

Any question, let me know: Wentao.bi@aptiv.com (248.892.3216)

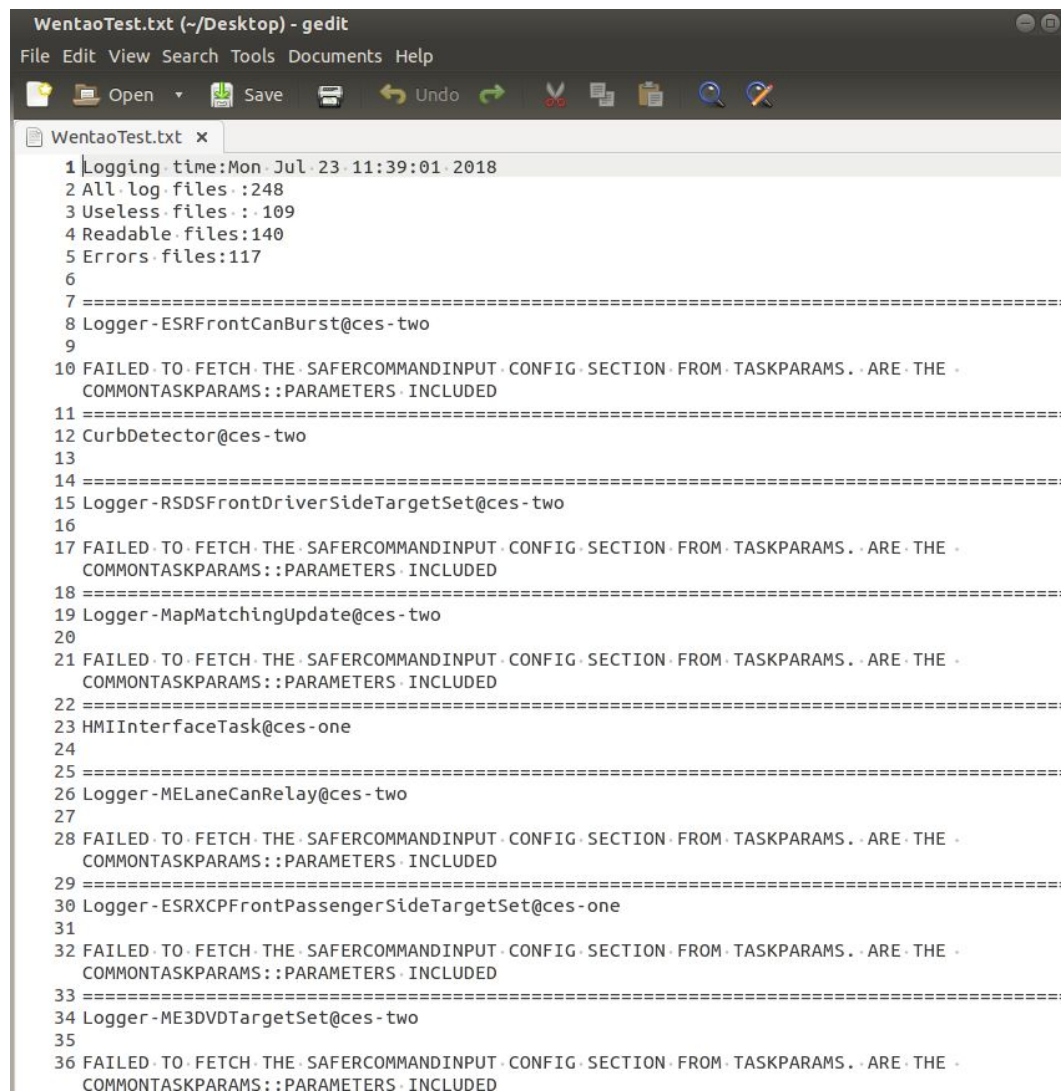
E.g. final Error file like this:

```
=====
Logger-SystemStatus@ces-oe

FAILED TO FETCH SAFERCOMMANDINPUT CONFIG SECTION...
ERROR MESSAGE...
ERROR MESSAGE...
ERROR MESSAGE...
=====

Logger-SystemStatu@ces-one

ERROR MESSAGE...
ERROR MESSAGE...
ERROR MESSAGE...
=====
```



```
WentaoTest.txt (~/Desktop) - gedit
File Edit View Search Tools Documents Help
Open Save Undo Redo
WentaoTest.txt x
1 Logging time: Mon Jul 23 11:39:01 2018
2 All log files: 248
3 Useless files: 109
4 Readable files: 140
5 Errors files: 117
6
7 =====
8 Logger-ESRFrontCanBurst@ces-two
9
10 FAILED TO FETCH THE SAFERCOMMANDINPUT CONFIG SECTION FROM TASKPARAMS. ARE THE
    COMMONTASKPARAMS::PARAMETERS INCLUDED
11 =====
12 CurbDetector@ces-two
13
14 =====
15 Logger-RSDSFrontDriverSideTargetSet@ces-two
16
17 FAILED TO FETCH THE SAFERCOMMANDINPUT CONFIG SECTION FROM TASKPARAMS. ARE THE
    COMMONTASKPARAMS::PARAMETERS INCLUDED
18 =====
19 Logger-MapMatchingUpdate@ces-two
20
21 FAILED TO FETCH THE SAFERCOMMANDINPUT CONFIG SECTION FROM TASKPARAMS. ARE THE
    COMMONTASKPARAMS::PARAMETERS INCLUDED
22 =====
23 HMIInterfaceTask@ces-one
24
25 =====
26 Logger-MELaneCanRelay@ces-two
27
28 FAILED TO FETCH THE SAFERCOMMANDINPUT CONFIG SECTION FROM TASKPARAMS. ARE THE
    COMMONTASKPARAMS::PARAMETERS INCLUDED
29 =====
30 Logger-ESRXCPFrontPassengerSideTargetSet@ces-one
31
32 FAILED TO FETCH THE SAFERCOMMANDINPUT CONFIG SECTION FROM TASKPARAMS. ARE THE
    COMMONTASKPARAMS::PARAMETERS INCLUDED
33 =====
34 Logger-ME3DVDTargetSet@ces-two
35
36 FAILED TO FETCH THE SAFERCOMMANDINPUT CONFIG SECTION FROM TASKPARAMS. ARE THE
    COMMONTASKPARAMS::PARAMETERS INCLUDED
```