

ENM PM Event File Decoder

User Guide

Copyright

© Ericsson AB 2017. All rights reserved. No part of this document may be reproduced in any form without the written permission of the copyright owner.

Disclaimer

The contents of this document are subject to revision without notice due to continued progress in methodology, design and manufacturing. Ericsson shall have no liability for any error or damage of any kind resulting from the use of this document.

Trademark List



Contents

1	ENM PM Event File Decoder	1
2	Log on to the Scripting Cluster	3
3	Using Event File Decoder	4
4	Decode a Single eNodeB DU/Baseband Radio Node or eNodeB Pico Radio Node CellTrace File	5
5	Decode a Single eNodeB DU/Baseband Radio Node or eNodeB Pico Radio Node UeTrace File	6
6	Decode a Single SGSN-MME UETrace File	7
7	Decode a Single SGSN-MME CTUM File	8
8	Decode a Single SGSN-MME EBM File	9
9	Decode a Single RNC GPEH file	10
10	Decode a Single RNC UETR file	11
11	Decode a Single RNC CTR file	12
12	Suppress the Compression of the Output File	13
13	Copy a Decoded File to Another Location	14
14	Run the Decoder with Help	15
15	Cleanup of Files that have exceeded Retention Time	16
16	Troubleshooting of PM Event File Decoder	17
16.1	Lock File Removal Fails	17
16.2	File System Use Exceeds Threshold	18
16.3	Unable to Find RNC GPEH Main File	18
16.4	Unable to determine name of RNC GPEH main file	19
16.5	Not Decoding RNC GPEH File	20
	References	21





1 ENM PM Event File Decoder

PM Event File Decoder (EFD) decodes trace files collected from Network Elements (NEs) by the Performance Management Initiation and Collection (PMIC) application.

The trace files collected from the NEs are primarily in compressed binary format. EFD is used to decode these binary files into a compressed or uncompressed plain-text `.csv` format.

Note: The exception is SGSN-MME Uetrace files which are in XML format and EFD decodes them into `pcap` format

The decoded files can be read by post-processing analysis tools (for example MS Excel), where the contents can be further analyzed

The supported trace types are:

- CellTrace files from eNodeB DU/Baseband Radio Node or eNodeB Pico Radio Node
- UETrace file from eNodeB DU/Baseband Radio Node or eNodeB Pico Radio Node or from SGSN-MME
- EBM file from SGSN-MME
- CTUM file from SGSN-MME
- GPEH, CTR and UETR files from RNC

This tool is not intended to be a Machine to Machine NBI and support issues related to such usage are not facilitated.

Requirements

- The ENM user has a connection to a General Scripting VM in ENM
- The ENM user has the correct role and/or capabilities to execute the EFD commands. The roles that can execute EFD commands are the 'Scripting_Operator' system role or a custom role that includes the 'Scripting Access CLI' capability together with one or more of the decoding capabilities from the table below.



Table 1

Capability	Description
Scripting NBI Decoding	Allows decoding of all trace types
Scripting LTE Celltrace Decoding	Allows decoding of LTE CELLTRACE files
Scripting LTE UE Trace Decoding	Allows decoding of LTE UETRACE files
Scripting MME CTUM Decoding	Allows decoding of MME CTUM files
Scripting MME EBM Decoding	Allows decoding of MME EBM files
Scripting MME UE Trace Decoding	Allows decoding of MME UETRACE files
Scripting RNC CTR Decoding	Allows decoding of RNC CTR files
Scripting RNC GPEH Decoding	Allows decoding of RNC GPEH files
Scripting RNC UETR Decoding	Allows decoding of RNC UETR files



2 Log on to the Scripting Cluster

Users must be logged on one of the General Scripting VMs (**scp-**-scripting**) using a Linux shell to use the PM Event File Decoder.

1. Connect to a General Scripting VM using the steps specified in the section "Log on to a Scripting KVM" or in the section "Load Balancing for AMOS and General Scripting VMs" in the *ENM System Operators Guide* [2].

Result

The user is now logged on to a General Scripting VM. A shell terminal is displayed with a prompt.

Example 1 EFDUser Logs on to scp-1-scripting

```
[another.linux.machine~]$ ssh -p 5023 EFDUser@enml234.abc.ericsson.se
EFDUser@scp-1-scripting's password:
[EFDUser@scp-1-scripting ~]$
```



3 Using Event File Decoder

When using Event File Decoder, the operator needs to provide the full path of the PM event file to decode.

Note: The location of the PM event files collected by ENM is determined by Performance Management, Initiation and Control (PMIC) feature. For more information on the configuration parameters used by PMIC for file storage, refer to PMIC File Collection - Write Data Files to NBI (North Bound Interface) in the *ENM System Administration Guide* [1]

The Event File Decoder program runs using a command appropriate to the trace type being decoded. For example `'lteCellTraceFileDecoder <full path to file>'` decodes a Cell Trace file.

All the commands with examples are given in the following chapters. EFD supports one user at a time decoding one file at a time

The output files are stored in the predefined location `/ericsson/tfd_cli_output/<TRACE_TYPE>/`. No other output directory can be defined.

Output files are compressed by default, with an option to suppress compression on a file by file basis. The output files can be copied to other locations using standard Linux commands.

Output files that have exceeded the configurable retention time are periodically deleted.

Also to prevent output files from filling the file system volume, there is a file system usage limit beyond which it is not possible to create new files.



4 Decode a Single eNodeB DU/Baseband Radio Node or eNodeB Pico Radio Node CellTrace File

Run the Event File Decoder program providing a valid encoded eNodeB DU/Baseband Radio Node or eNodeB Pico Radio Node CellTrace file as input.

The resulting output file has the same filename as the input file, but with .csv.gz extension.

For example, if the input file is

```
A20151009.1045-1100_SubNetwork=LTE,MeContext=ENB512_cellt  
racefile_DUL1_1.bin.gz, the output file is
```

```
A20151009.1045-1100_SubNetwork=LTE,MeContext=ENB512_cellt  
racefile_DUL1_1.csv.gz.
```

The output file is stored in the output directory (/ericsson/tfd_cli_output/LTE_CELLTRACE).

Example 2 EFDUser Decodes LTE CellTrace File

```
[EFDUser@scp-1-scripting ~]$ lteCellTraceFileDecoder /ericsson/pmic1/CELLTRAC →  
E/SubNetwork=Utran6_R,SubNetwork=LTE,MeContext=ENB512/A20151009.1045-1100_Sub →  
Network=LTE,MeContext=ENB512_celltracefile_DUL1_1.bin.gz →  
Decoding file: →  
A20151009.1045-1100_SubNetwork=LTE,MeContext=ENB512_celltracefile_DUL1_1.bin. →  
gz →  
Done. Output file is /ericsson/tfd_cli_output/LTE_CELLTRACE/A20151009.1045-11 →  
00_SubNetwork=LTE,MeContext=ENB512_celltracefile_DUL1_1.csv.gz →  
[EFDUser@scp-1-scripting ~]$
```



5 Decode a Single eNodeB DU/Baseband Radio Node or eNodeB Pico Radio Node UeTrace File

Run the Event File Decoder program providing a valid encoded eNodeB DU/Baseband Radio Node or eNodeB Pico Radio Node UeTrace file as input.

The resulting output file has the same filename as the input file, but with .csv.gz extension.

For example, if the input file is

```
A20160308.1315+0000-1330+0000_SubNetwork=LTESubNetwork,MeContext=LTE01ERBS00057_123456789ABC_123456789ABCDEF_uetracefile.bin.gz, the output file is
```

```
A20160308.1315+0000-1330+0000_SubNetwork=LTESubNetwork,MeContext=LTE01ERBS00057_123456789ABC_123456789ABCDEF_uetracefile.csv.gz
```

The output file is stored in the output directory (/ericsson/tfd_cli_output/LTE_UETRACE).

Example 3 EFDUser Decodes LTE UETrace File

```
[EFDUser@scp-1-scripting ~]$ lteUeTraceFileDecoder /ericsson/pmic1/UETRACE/ERBS/123456789ABC/A20160308.1315+0000-1330+0000_SubNetwork=LTESubNetwork,MeContext=LTE01ERBS00057_123456789ABC_123456789ABCDEF_uetracefile.bin.gz →
Decoding file: →
A20160308.1315+0000-1330+0000_SubNetwork=LTESubNetwork,MeContext=LTE01ERBS00057_123456789ABC_123456789ABCDEF_uetracefile.bin.gz →
Done. Output file is /ericsson/tfd_cli_output/LTE_UETRACE/A20160308.1315+0000-1330+0000_SubNetwork=LTESubNetwork,MeContext=LTE01ERBS00057_123456789ABC_123456789ABCDEF_uetracefile.csv.gz →
[EFDUser@scp-1-scripting ~]$
```



6 Decode a Single SGSN-MME UETrace File

Run the Event File Decoder program providing a valid encoded SGSN-MME UeTrace file as input.

The resulting output filename is `ue_trace<IMSI>.pcap`.

For example, if the input file is

`B20120319.0703+0100-20110920.1830+0200-MME.target.FIV1._1_ue_trace.19.xml`, the output file is `ue_trace.IMSI123789000000111.pcap`.

The output file is stored in the output directory (`/ericsson/tfd_cli_output/MME_UETRACE`).

Example 4 EFDUser Decodes SGSN-MME UeTrace File

```
[EFDUser@scp-1-scripting ~]$ mmeUeTraceFileDecoder /ericsson/pmic1/UETRACE/MME/MeContext=SGSN-15B-V601,ManagedElement=SGSN-15B-v603/B20120319.0703+0100-20110920.1830+0200-MME.target.FIV1._1_ue_trace.19.xml →
Decoding file: →
A20160308.1315+0000-1330+0000 SubNetwork=LTESubNetwork,MeContext=LTE01ERBS000 →
57 123456789ABC 123456789ABCDEF uetracefile.bin.gz →
Processing:/ericsson/pmic1/UETRACE/MME/MeContext=SGSN-15B-V601,ManagedElement →
=SGSN-15B-v603/B20120319.0703+0100-20110920.1830+0200-MME.target.FIV1._1_ue_t →
race.19.xml →
Result:IMSI 123789000000111 : Written to ue_trace.IMSI123789000000111.pcap →
Done. Output file is /ericsson/tfd_cli_output/MME_UETRACE/ue_trace.IMSI123789 →
000000111.pcap →
[EFDUser@scp-1-scripting ~]$
```



7 Decode a Single SGSN-MME CTUM File

Run the Event File Decoder program providing a valid encoded SGSN-MME CTUM file as input.

The resulting output file has the same filename as the input file, but with .csv.gz extension.

For example, if the input file is

```
A20141014.1040+0800-20141014.1045+0800_1_ctum-1413283846629.1.gz, the output file is  
A20141014.1040+0800-20141014.1045+0800_1_ctum-1413283846629.1.csv.gz .
```

The output file is stored in the output directory (/ericsson/tfd_cli_output/MME_CTUM).

Example 5 EFDUser Decodes SGSN-MME CTUM File

```
[EFDUser@scp-1-scripting ~]$ mmeCtumFileDecoder /ericsson/pmic1/CTUM/A20141014.1040+0800-20141014.1045+0800_1_ctum-1413283846629.1 →  
Decoding file:  
A20141014.1040+0800-20141014.1045+0800_1_ctum-1413283846629.1.gz  
Done. Output file is /ericsson/tfd_cli_output/MME_CTUM/A20141014.1040+0800-20141014.1045+0800_1_ctum-1413283846629.1.csv.gz →  
[EFDUser@scp-1-scripting ~]$
```



8 Decode a Single SGSN-MME EBM File

Run the Event File Decoder program providing a valid encoded SGSN-MME EBM file as input.

The resulting output file has the same filename as the input file, but with .csv.gz extension.

For example, if the input file is

A20160817.0638+0000-20160817.0639+0000_1_ebs.1508, the output file is A20160817.0638+0000-20160817.0639+0000_1_ebs.1508.csv.gz.

The output file is stored in the output directory (/ericsson/tfd_cli_output/MME_EBM).

Example 6 EFDUser Decodes SGSN-MME EBM File

```
[EFDUser@scp-1-scripting ~]$ mmeEbmFileDecoder /ericsson/pmic1/ebm/data/ManagedElement=SGSN-15B-V626/A20160817.0638+0000-20160817.0639+0000_1_ebs.1508 →
Decoding file:
A20160817.0638+0000-20160817.0639+0000_1_ebs.1508
Done. Output file is /ericsson/tfd_cli_output/MME_EBM/A20160817.0638+0000-20160817.0639+0000_1_ebs.1508.csv.gz →
[EFDUser@scp-1-scripting ~]$
```



9 Decode a Single RNC GPEH file

For GPEH recording the RNC produces a main file and sub files. The main file includes metadata information while the sub files contain the event data. When PMIC collects the GPEH files it adds "Mp0" to the name of the GPEH main file and "Mp<n>" to the name of the sub files where n has value 1,2,3... PMIC stores all the files (main file and sub files) for the same ROP in the same directory.

EFD can be used to decode the event data from the GPEH sub files. While decoding the GPEH sub file, EFD uses the corresponding GPEH main file to determine the version of the data. To do this, EFD looks for the corresponding Mp0 file in the same directory as the specified GPEH sub file. If it cannot find the Mp0 file, it will be unable to decode the GPEH sub file.

Steps

Run the Event File Decoder program providing a valid encoded RNC GPEH file as input.

The resulting output file has the same filename as the input file, but with `.csv.gz` extension.

For example, if the input file is

```
A20160826.1115+0000-1130+0000_SubNetwork=RNC1,MeContext=RNC1_rnc_gpehfile_Mp1.bin.gz, the output file is
```

```
A20160826.1115+0000-1130+0000_SubNetwork=RNC1,MeContext=RNC1_rnc_gpehfile_Mp1.csv.gz.
```

The output file is stored in the output directory (`/ericsson/tfd_cli_output/RNC_GPEH`).

Example 7 EFDUser Decodes RNC GPEH File

```
[EFDUser@scp-1-scripting ~]$ decodeRncGpehFile /ericsson/pmic1/GPEH/MeContext →
=RNC1/A20160826.1115+0000-1130+0000_SubNetwork=RNC1,MeContext=RNC1_rnc_gpehfi →
le_Mp1.bin.gz
Decoding file:
A20160826.1115+0000-1130+0000_SubNetwork=RNC1,MeContext=RNC1_rnc_gpehfile_Mp1 →
.bin.gz
Done. Output file is /ericsson/tfd_cli_output/RNC_GPEH/A20160826.1115+0000-11 →
30+0000_SubNetwork=RNC1,MeContext=RNC1_rnc_gpehfile_Mp1.csv.gz
[EFDUser@scp-1-scripting ~]$
```

Note: For RNC GPEH files, EFD decodes the ASN.1 data from all External events and the INTERNAL_SIB11_NCELL_INFO internal event.



10 Decode a Single RNC UETR file

Run the Event File Decoder program providing a valid encoded RNC UETR file as input.

The resulting output file has the same filename as the input file, but with `.csv.gz` extension.

For example, if the input file is

`A20170616.1345+0100-1400+0100_MeContext=RNC1_10006_123456000000001_uetrfile.bin.gz`, the output file is `A20170616.1345+0100-1400+0100_MeContext=RNC1_10006_123456000000001_uetrfile.csv.gz`

The output file is stored in the output directory (`/ericsson/tfd_cli_output/RNC_UETR.`)

Example 8 EFDUser Decodes RNC UETR File

```
[EFDUser@scp-1-scripting ~]$ decodeRncUetrFile /ericsson/pmic1/UETR/MeContext →
=RNC1/A20170616.1345+0100-1400+0100_MeContext=RNC1_10006_123456000000001_uetr →
file.bin.gz
Decoding file:
A20170616.1345+0100-1400+0100_MeContext=RNC1_10006_123456000000001_uetrfile.b →
in.gz
Done. Output file is /ericsson/tfd_cli_output/RNC_UETR/A20170616.1345+0100-14 →
00+0100_MeContext=RNC1_10006_123456000000001_uetrfile.csv.gz
[EFDUser@scp-1-scripting ~]$
```

Note: For RNC GPEH files, EFD decodes the ASN.1 data from all External events and the INTERNAL_MEAS internal event.



11 Decode a Single RNC CTR file

Run the Event File Decoder program providing a valid encoded RNC CTR file as input.

The resulting output file has the same filename as the input file, but with `.csv.gz` extension.

For example, if the input file is

```
A20170511.1845+0000-1900+0000_MeContext=RNC1_20001_MeContext=RNC1,ManagedElement=1,RncFunction=1,UtranCell=RNC1-1-1_ctrfile.bin.gz, the output file is
```

```
A20170511.1845+0000-1900+0000_MeContext=RNC1_20001_MeContext=RNC1,ManagedElement=1,RncFunction=1,UtranCell=RNC1-1-1_ctrfile.csv.gz
```

The output file is stored in the output directory (`/ericsson/tfd_cli_output/RNC_CTR`)

Example 9 EFDUser Decodes RNC CTRFile

```
[EFDUser@scp-1-scripting ~]$ decodeRncUetrFile /ericsson/pmic1/UETR/MeContext →
=RNC1/A20170616.1345+0100-1400+0100_MeContext=RNC1_10006_123456000000001_uetr →
file.bin.gz →
Decoding file: →
A20170616.1345+0100-1400+0100_MeContext=RNC1_10006_123456000000001_uetrfile.b →
in.gz →
Done. Output file is /ericsson/tfd_cli_output/RNC_UETR/A20170616.1345+0100-14 →
00+0100_MeContext=RNC1_10006_123456000000001_uetrfile.csv.gz →
[EFDUser@scp-1-scripting ~]$
```

Note: For RNC GPEH files, EFD decodes the ASN.1 data from all External events and the INTERNAL_MEAS internal event.



12 Suppress the Compression of the Output File

On a file basis, it is possible to suppress the compression of the output file. This option only applies to the current command and is not a system-wide setting.

Suppress the compression of the output file by adding the `-nc` or `--no_compression` options to the command.

Example 10 Suppress the Compression with `-nc`

```
[EFDUser@scp-1-scripting ~]$ lteUeTraceFileDecoder /ericsson/pmic1/UETRACE/ERBS/123456789ABC/A20160308.1315+0000-1330+0000_SubNetwork=LTESubNetwork,MeContext=LTE01ERBS00057_123456789ABC_123456789ABCDEF_uetracefile.bin.gz -nc →
Decoding file: →
A20160308.1315+0000-1330+0000_SubNetwork=LTESubNetwork,MeContext=LTE01ERBS00057_123456789ABC_123456789ABCDEF_uetracefile.bin.gz →
Done. Output file is /ericsson/tfd_cli_output/LTE_UETRACE/A20160308.1315+0000-1330+0000_SubNetwork=LTESubNetwork,MeContext=LTE01ERBS00057_123456789ABC_123456789ABCDEF_uetracefile.csv →
[EFDUser@scp-1-scripting ~]$
```

Example 11 Suppress the Compression with `--no_compression`

```
[EFDUser@scp-1-scripting ~]$ lteCellTraceFileDecoder /ericsson/pmic1/CELLTRACE/SubNetwork=Utran6_R,SubNetwork=LTE,MeContext=ENB512/A20151009.1045-1100_SubNetwork=LTE,MeContext=ENB512_celltracefile_DUL1_1.bin.gz --no_compression →
Decoding file: →
A20151009.1045-1100_SubNetwork=LTE,MeContext=ENB512_celltracefile_DUL1_1.bin.gz →
Done. Output file is /ericsson/tfd_cli_output/LTE_CELLTRACE/A20151009.1045-1100_SubNetwork=LTE,MeContext=ENB512_celltracefile_DUL1_1.csv →
[EFDUser@scp-1-scripting ~]$
```



13 Copy a Decoded File to Another Location

The decoded files can be copied within ENM using standard Linux commands, or outside ENM using SFTP.

1. Copy the file from the default output directory to the current working directory with the `cp` Linux command.
2. Open an SFTP session and copy the file to an external location (as defined by `<IP address>`).

Example 12

EFDUser default output directory

aUser external location

```
[EFDUser@scp-1-scripting ~]$ cp /ericsson/tfd_cli_output/LTE_CELLTRACE/A20151009.1045-1100_SubNetwork=LTE,MeContext=ENB512_celltracefile_DUL1_1.csv.gz .
[EFDUser@scp-1-scripting ~]$ sftp aUser@<IP address>
Connecting to <IP address>...
aUser@<IP address>'s password:
sftp> pwd
Remote working directory: /aUser
sftp> lpwd
Local working directory: /home/shared/EFDUser
sftp> put A20151009.1045-1100_SubNetwork=LTE,MeContext=ENB512_celltracefile_DUL1_1.csv.gz
Uploading A20151009.1045-1100_SubNetwork=LTE,MeContext=ENB512_celltracefile_DUL1_1.csv.gz to /aUser/A20151009.1045-1100_SubNetwork=LTE,MeContext=ENB512_celltracefile_DUL1_1.csv.gz
A20151009.1045-1100_SubNetwork=LTE,MeContext=ENB512_celltracefile_DUL1_1.csv.gz 1
00% 3607KB 1.8MB/s 00:02
sftp> exit
[EFDUser@scp-1-scripting ~]$
```



14 Run the Decoder with Help

Run the Event File Decoder program with the `-h` (`--help`) option.

Example 13 EFDUser for lteCellTraceFileDecoder

```
[EFDUser@scp-1-scripting ~]$ lteUeTraceFileDecoder -h
usage: lteUeTraceFileDecoder [-h] [--type [TYPE]] [file_name]
trace file decoder -- decodes trace files to either a readable csv format for
celltrace & ue trace or pcap format for mme uetrace

positional arguments:
  file_name          specify full path and name of the file to be decoded

optional arguments:
  -h, --help        show this help message and exit
  --type [TYPE]     specify type of the file to be decoded, valid types are
                    LTE_CELLTRACE, LTE_UETRACE

Full path to file must be specified Output is placed in 'test_out' directory.
The disk usage volume is checked prior to decoding files and if the threshold
is exceeded, no new files will be decoded until old files are deleted.
Decoder will skip already decoded files. File is considered decoded if a file
with the same name and applicable extension exists in the output directory.
```



15 Cleanup of Files that have exceeded Retention Time

EFD periodically deletes all decoded files in the output directory that have exceeded the retention time.

The retention time of the decoded files is configurable.

For more information on the configuration parameter, refer to the *Pm Event File Decoder Administration Tasks* in the *ENM Performance Management System Administration Guide* [1]



16 Troubleshooting of PM Event File Decoder

The section provides troubleshooting steps recommended to diagnose and fix common problems with PM Event File Decoder (EFD)

16.1 Lock File Removal Fails

Remove the lock file manually.

While running the Event File Decoder, the following message is displayed:

```
Lockfile exists, but is older than 5 minutes. Remove the →
/ericsson/tfd_cli_output/.lock file manually to continue.
```

Cause

The Event File Decoder implements a lock file mechanism which prevents several users from running at the same time. If the program is interrupted it can fail to remove the lockfile located in the default output directory before terminating.

Solution

Responsible party: EFDUser (example)

Locate and delete the lockfile (named `.lock`).

```
[EFDUser@scp-1-scripting ~]$ lteCellTraceFileDecoder /ericsson/pmic1/CELLTRAC →
E/SubNetwork=Utran6_R,SubNetwork=LTE,MeContext=ENB512/A20151009.1045-1100_Sub →
Network=LTE,MeContext=ENB512_celltracefile_DUL1_1.bin.gz →
Lockfile exists, but is older than 5 minutes. →
Remove the /ericsson/tfd_cli_output/.lock file manually to continue. →
[EFDUser@scp-1-scripting ~]$ →
[EFDUser@scp-1-scripting ~]$ →
[EFDUser@scp-1-scripting ~]$ ls -lA /ericsson/tfd_cli_output/ →
total 93 →
-rw-r--r--. 1 EFDUser      scripting_decoder      0 Mar 18 09:56 .lock →
[EFDUser@scp-1-scripting ~]$ rm /ericsson/tfd_cli_output/.lock →
[EFDUser@scp-1-scripting ~]$ lteCellTraceFileDecoder /ericsson/pmic1/CELLTRAC →
E/SubNetwork=Utran6_R,SubNetwork=LTE,MeContext=ENB512/A20151009.1045-1100_Sub →
Network=LTE,MeContext=ENB512_celltracefile_DUL1_1.bin.gz →
Decoding file: →
A20151009.1045-1100_SubNetwork=LTE,MeContext=ENB512_celltracefile_DUL1_1.bin. →
gzz →
Done. Output directory is /ericsson/tfd_cli_output/LTE_CELLTRACE/ →
[EFDUser@scp-1-scripting ~]$
```



16.2 File System Use Exceeds Threshold

Bring the file system use below 85%.

While running the Event File Decoder, the following message is displayed:

```
Disk usage in /ericsson/tfd_cli_output/ exceeds the threshold of 85%.
```

Cause

To ensure that the output volume does not become locked, requiring an admin user to unlock, the Event File Decoder program stops decoding files if the file system usage in the output volume exceeds the 85% usage threshold.

Solution

Responsible party: EFDUser (example)

Remove old decoded files.

```
[EFDUser@scp-1-scripting ~]$ lteCellTraceFileDecoder /ericsson/pmic1/CELLTRAC →
E/SubNetwork=Utran6_R,SubNetwork=LTE,MeContext=ENB512/A20151009.1045-1100_Sub →
Network=LTE,MeContext=ENB512_celltracefile_DUL1_1.bin.gz →
Disk usage in /ericsson/tfd_cli_output/ exceeds the threshold of 85% →
[EFDUser@scp-1-scripting ~]$ rm /ericsson/tfd_cli_output/LTE_CELLTRACE/A20151 →
009.1030-1045_SubNetwork=LTE,MeContext=ENB512_celltracefile_DUL1_1.csv.gz →
[EFDUser@scp-1-scripting ~]$ →
[EFDUser@scp-1-scripting ~]$ lteCellTraceFileDecoder /ericsson/pmic1/CELLTRAC →
E/SubNetwork=Utran6_R,SubNetwork=LTE,MeContext=ENB512/A20151009.1045-1100_Sub →
Network=LTE,MeContext=ENB512_celltracefile_DUL1_1.bin.gz →
Disk usage in /ericsson/tfd_cli_output/ exceeds the threshold of 85% →
[EFDUser@scp-1-scripting ~]$ →
[EFDUser@scp-1-scripting ~]$ rm /ericsson/tfd_cli_output/LTE_CELLTRACE/A20151 →
009.1015-1030_SubNetwork=LTE,MeContext=ENB512_celltracefile_DUL1_1.csv.gz →
[EFDUser@scp-1-scripting ~]$ lteCellTraceFileDecoder /ericsson/pmic1/CELLTRAC →
E/SubNetwork=Utran6_R,SubNetwork=LTE,MeContext=ENB512/A20151009.1045-1100_Sub →
Network=LTE,MeContext=ENB512_celltracefile_DUL1_1.bin.gz →
Disk usage in /ericsson/tfd_cli_output/ exceeds the threshold of 85% →
[EFDUser@scp-1-scripting ~]$ →
[EFDUser@scp-1-scripting ~]$ rm /ericsson/tfd_cli_output/LTE_CELLTRACE/A20151 →
009.1000-1015_SubNetwork=LTE,MeContext=ENB512_celltracefile_DUL1_1.csv.gz →
[EFDUser@scp-1-scripting ~]$ lteCellTraceFileDecoder /ericsson/pmic1/CELLTRAC →
E/SubNetwork=Utran6_R,SubNetwork=LTE,MeContext=ENB512/A20151009.1045-1100_Sub →
Network=LTE,MeContext=ENB512_celltracefile_DUL1_1.bin.gz →
Decoding file: →
A20151009.1045-1100_SubNetwork=LTE,MeContext=ENB512_celltracefile_DUL1_1.bin. →
gz →
Done. Output directory is /ericsson/tfd_cli_output/LTE_UETRACE →
[EFDUser@scp-1-scripting ~]$
```

16.3 Unable to Find RNC GPEH Main File

While trying to decode an RNC GPEH file, the following message is displayed:

```
Unable to find GPEH main file (Mp0 file) in directory <location of GPEH file to →
be decoded>.
```



Cause

As described in the section *Decode a Single RNC GPEH file* [Decode a Single RNC GPEH file](#) on page 10, when decoding a GPEH sub file, EFD uses the corresponding GPEH main file to determine the version of the data. In this case EFD does not find the corresponding GPEH main file.

Solution

Responsible party: EFDUser (example)

If the GPEH main file is not in the same directory as the relevant GPEH subfile, then copy the GPEH main file to the same directory as the relevant GPEH subfile.

If the main file is in the same directory as the GPEH subfile, then ensure that the GPEH main file follows the ENM naming convention for collected GPEH files where Mp0 is added to the name of the GPEH main file.

Subfile name:

```
A20160826.1115+0000-1130+0000_SubNetwork=RNC1,MeContext= →
RNC1_rnc_gpehfile_Mp1.bin.gz
```

Corresponding main file name:

```
A20160826.1115+0000-1130+0000_SubNetwork=RNC1,MeContext= →
RNC1_rnc_gpehfile_Mp0.bin.gz
```

16.4 Unable to determine name of RNC GPEH main file

While trying to decode an RNC GPEH file, the following message is displayed:

```
Unable to determine name of GPEH main file (Mp0 file) from input filename <name →
of GPEH file to be decoded>
```

Cause

As described in the section *Decode a Single RNC GPEH file* [Decode a Single RNC GPEH file](#) on page 10, when decoding a GPEH sub file, EFD uses the corresponding GPEH main file to determine the version of the data. To do this, EFD looks for the corresponding Mp0 file in the same directory as the specified GPEH sub file.



If it cannot find the Mp0 file, it will be unable to decode the GPEH sub file. EFD uses the ENM naming convention of collected GPEH sub files to determine the corresponding GPEH main file.

In this case the GPEH sub file to be decoded does not follow the ENM naming convention where the Mp number is added to the file name.

Subfile name:

```
A20160826.1115+0000-1130+0000_SubNetwork=RNC1,MeContext=RNC1_rnc_gpehfile_Mp1.bin.gz →
```

Corresponding main file name:

```
A20160826.1115+0000-1130+0000_SubNetwork=RNC1,MeContext=RNC1_rnc_gpehfile_Mp0.bin.gz →
```

Solution

Responsible party: EFDUser (example)

Rename the GPEH sub file to follow the ENM naming convention for collected GPEH files

16.5 Not Decoding RNC GPEH File

While trying to decode an RNC GPEH file, the following message is displayed:

```
Not decoding file <file to be decoded> as it is the GPEH main file (Mp0 file)
```

Cause

As described in the section *Decode a Single RNC GPEH file* [Decode a Single RNC GPEH file](#) on page 10, for GPEH recording the RNC produces a main file and sub files. The main file includes metadata information while the sub files contain the event data. EFD only decodes files containing event data and hence only decodes GPEH sub files.

Solution

None. EFD does not decode GPEH main files.



References

Ref.		
[1]	<i>ENM Performance System Administration Guide</i> 1/1543-AOM 901 151-3 Available in ENM CPI Library	
[2]	<i>ENM Operators Guide</i> 1/1553-AOM 901 151 Available in ENM CPI Library	