

MTAS Troubleshooting Guideline

MTAS

TROUBLESHOOTING

Copyright

© Ericsson AB 2016. 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

All trademarks mentioned herein are the property of their respective owners. These are shown in the document Trademark Information.



Contents

1	Introduction	1
1.1	Prerequisites	1
2	Tools	3
2.1	Information Retrieval	3
2.2	Cluster Status Tools	4
2.3	NetTrace	4
3	Troubleshooting Functions	7
3.1	Alarms	7
3.2	Logging	7
3.3	Application Specific Logs	8
3.4	Tracing	9
3.5	Counters	11
3.6	Software Level Checks	11
3.7	Handle Log Files and Counter Files	12
3.8	Change Syslog Level	12
3.9	Restart	13
3.10	Recover H.248 Link	13
3.11	Sessions Hanging	14
4	Troubleshooting Procedure	17
4.1	Perform Node Health Check	17
4.2	Check Alarms	17
4.3	Verify Node Configuration	18
4.4	Troubleshoot Scale in Failures	19
4.5	Troubleshoot License Manager	21
5	Application Recovery Procedure	23
5.1	Diameter Interface	23
5.2	Handle Hung Carrier Select Synchronization	26
6	Trouble Report	29





1 Introduction

This document describes how to perform the troubleshooting procedure in the MTAS.

1.1 Prerequisites

This section describes the prerequisites for this document.

It is assumed that users of this document are familiar with performing operations within the area for O&M in general.

1.1.1 Documents

Before starting this procedure, ensure that the following documents have been read:

- *MTAS AppTrace*
- *MTAS Fault Codes Catalogue*
- *MTAS Health Check*
- *MTAS Network Tracing*
- *MTAS Scaling Management Guide*

1.1.2 Conditions

Certain troubleshooting activities can have an impact on node performance. For example, trace or log activation can be traffic disturbing and is not recommended without first consulting Ericsson.





2 Tools

This section describes the tools that can be used to troubleshoot the MTAS.

2.1 Information Retrieval

vDicos related information retrieval is to be performed with the help of the vDicos Command Line Utility (CLU). This utility operates on those parts of the vDicos MIM (stored in IMM) that are not made public through COM and requires in general some predefined logic in manipulating Managed Object (MO) attributes and related values.

The CLU is a “system global” utility meaning that it can be started on any node the vDicos cluster is loaded on, and can operate on all the nodes of the vDicos cluster independently of the node the user is logged in and the required CLU tool is started on. This also means, certain CLU tools can be executed on more than one node in the vDicos cluster.

Therefore, to scope by need a certain CLU tool the user is provided with the following possibilities:

- Scoping on node level.

As certain tools can be executed on more than one node, in the vDicos cluster the CLU user is allowed to specify the node or nodes the certain tool is to be executed on. This is called “node level scoping”. If no “node level scoping” is specified by the user, the tool is executed by CLU on all the nodes the started tool is available on.

- Scoping on execution domain level.

Each CLU tool is associated (design time or runtime) with a certain “execution domain” which represents a hierarchical organization, the “region of execution”, of the relevant tool. For instance, an “execution domain” can refer to a vDicos Virtual Machine (VM) that the tool is to be executed on, a certain object that the tool is associated with, and so on. The “execution domain”, for example, “lpmsv.agent.vm0”, is specified as a dotted sequence of strings, like a namespace, where the more generic domain sits on front place (left side), while the more specific domain sits on the back (right side). A CLU user can use both scoping methods to properly scope the execution of the started tool. If no scoping is specified, the started tool is executed on all the nodes for all the valid “execution domains”.



2.1.1 Alarms and Notifications

The COM FM (Fault Management) allows the operator to detect faults and malfunctions on the system. Based on the information carried in the notifications, the operator can locate the source of the event and the relevant documentation. Based on this information, actions can be taken to resolve or prevent failures. Refer to *MTAS Alarm List* and *Fault Management*.

2.1.2 SS7 Configuration

The Signaling Manager provides a Graphical User Interface (GUI) and a Command-Line Interface (CLI) for the configuration and operation of the signaling stack.

2.2 Cluster Status Tools

This section describes the tools that can be used to check the cluster status.

2.2.1 cmw-status

CoreMW provides a tool to verify the cluster health status `cmw-status`. When using this tool only failing items are printed, unless the `-v` flag is specified.

2.2.2 cdsv-get-user-state

The `cdsv-get-user-state` tool prints the CDSv user state.

2.2.3 cdsv-get-state

The `cdsv-get-state` tool prints the CDSv user state.

2.2.4 cdsv-get-node-state

The `cdsv-get-node-state` tool prints the state of the nodes.

2.3 NetTrace

Note: The inherent problem with observing the behavior of a system by tracing, is the consumed capacity of the tracing itself. If the cost is too high, it can interfere with the primary function of the system, at worst even causing system failure.

NetTrace is a tool that allows the user to trace SIP transactions that traverse the MTAS depending on user-defined filter criteria. These transactions are



formatted and output in standardized XML file format, according to the following 3GPP specification, refer to [3GPP TS 32.423, 8.1.0: Telecommunication management; Subscriber and equipment trace; Trace data definition and management](#).

This feature can be used for troubleshooting by customer support centers, during interoperability testing, and so on, with the benefit of improved total cost of ownership (TCO).

SIP transactions can be traced at two levels, min (minimum) and max (maximum).

Tracing is performed on all MTAS implemented SIP interfaces. However, no SIP interfaces can be filtered out.

vDicos AppTrace is used to realize NetTrace. For more information about NetTrace and AppTrace, refer to *MTAS Network Tracing* and *MTAS AppTrace*.





3 Troubleshooting Functions

This section describes the troubleshooting functions included in this guide.

3.1 Alarms

The COM FM (Fault Management) allows the operator to detect faults and malfunctions on the system. Based on the information carried in the alarms, the operator can locate the source of the event and the relevant documentation. Based on this information action can be taken to resolve or prevent failures. Refer to *MTAS Alarm List* and *Fault Management*.

3.2 Logging

This section describes the Event Logs for the node. These logs are to be provided to the next level of maintenance support in case the problem cannot be solved with the help of this instruction.

The files can contain information from a crash. If a crash has occurred, the files with information about the crash are collected in a tar file.

For more information about general log information, refer to *Data Collection Guideline for MTAS* and *MTAS Logs*.

The following error log file types exist:

- Syslog
- Processor logs
- Crashcollector

3.2.1 Syslog

The Syslog is a log file containing information about the main events in the processors. Examples of these events are processor disturbances and capsule abortions.

They are located in the `/var/log/<node_name>/messages` files.

For more information about the log file, refer to *MTAS Logs*.



3.2.2 Processor Logs

The processor logs contain processor execution printouts for each processor. These logs are found in the directory, `/cluster/storage/no-backup/cdclsv/log/lpmsv`. The processor logs are used by both CBA and the MTAS to include log information for the processors. The MTAS includes an Event History in the processor logs after a Process Abortion.

For more information about logging, refer to *MTAS Logs*.

MTAS AppTrace ends up in the processor logs. For more details about AppTrace, refer to *MTAS AppTrace*.

Default Trace ends up in the processor logs. For more details about Default Trace, refer to *MTAS AppTrace*.

3.2.3 Crashcollector

The `Crashcollector` files are collected (and generated) from the system and are then compressed and stored in the directory `/opt/cdclsv/storage/dumps/` with the name `logdump-<date>-<time>.<msc>.tgz`. The file contains the Capsule Abortion dumps and the related log files. Dump archives (contains the following directories) are generated periodically and can contain several Capsule Abortion dumps.

- `/opt/cdclsv/storage/autocc` Directory

The `autocc` directory contains the `vDicos` process abortion dump files and `coredump` files, and the result of the automated crash processing (backtrace) as a text file for each dump. The `vDicos` process abortion dumps use the following naming pattern: `vdicosvmca-<processorname>-<date>.<time>`. The related backtrace file has the same name, except it has `.txt` extension.

For more information about the error dumps, refer to *MTAS Error Dump*.

3.3 Application Specific Logs

This section describes the application-specific logs.

3.3.1 MTAS Applog

The MTAS applog log files contain information about terminated MTAS processes. Such a terminated process is a fault indication.

The `vDicosApplog` is stored in `/cluster/storage/no-backup/coremw/var/log/saflog/MTASAppLogs/vdicos` directory and for each application. MTAS log files specifically are stored with the following naming convention: `MTAS_<timestamp>.log`



For more information about the MTAS applog, refer to *Data Collection Guideline for MTAS* and *MTAS Logs*.

For more information about AppTrace, refer to *MTAS AppTrace*.

3.3.2 Catalina Log

The MTAS Catalina log files contain information about MTAS XDMS processes. Each Catalina file has a maximum size of 10 MB. Once this size has been reached, a new file is created. The total number of files can be up to 11, named `catalina.log`, `catalina.log.1`, and so on. The resulting string starts with the date, time, thread name, log level, AppTrace domain, session, and sequence number. Next part of the string is the “-” character, after which is the logging information. The resulting string ends up in the Catalina log and is printed to the following directory:

```
/opt/mmas/appserver/traffic_instance0/standalone/logs/
```

The following are examples of a Catalina string:

```
2015-09-15 10:10:01,260 [http-apr-127.0.0.1-8095-exec-8]  
DEBUG mtas.xdms.xdms.cai3g A192D168D83D100Z1441384753S516P  
65537 1688257881 - Document pre-edit
```

```
2015-09-15 10:07:07,879 [http-apr-127.0.0.1-8090-exec-2]  
DEBUG mtas.xdms.xdms.validation 127.0.0.1 simservs  
sip:user@telcom.com - ...application validated.
```

3.4 Tracing

This section describes how to use tracing.

3.4.1 NetTrace

The principle of NetTrace is to allow a user to observe and log SIP transactions traversing the MTAS for fault finding/localization purposes.

It is possible to trace the requests or responses sent and received by a particular `PublicId` or several `PublicIds`, or to only trace requests or responses where a particular `PublicId` is the originator or receiver. It is also possible to trace requests or responses between a specific originating and a terminating user.

There are two levels of tracing:

- Minimum level, where only a subset of the SIP headers is traced.
- Maximum level, where the complete SIP message is traced in hexadecimal format.



Note: Currently only the Minimum level is available.

3.4.1.1 Starting NetTrace Using AppTrace

For information about to start NetTrace using AppTrace, refer to *MTAS Network Tracing*.

3.4.2 Trace Profiles

This section describes how to use the preconfigured trace profiles.

To start the trace profiles:

1. Install a normal product (without static trace).
2. Log on to SC-1 or SC-2 (SC-2 in the example), using SSH and the root privilege.

```
ssh <address><port>
SC-2 login: root
Password:
```

3. From the SC, connect to one of the PLs using SSH, for example to PL-3:

```
SC-2:~ # ssh PL-3
```

4. List all trace profiles:

```
$ /opt/ericsson/cmco/traceprofiles/bin/imstrace list
```

For information about preconfigured trace profiles, refer to *MTAS AppTrace*.

5. Use the built-in help to get more information:

```
$ /opt/ericsson/cmco/traceprofiles/bin/imstrace help
<traceprofilename>
```

6. Start `imstrace` (that is, turn on the dynamic trace) with the chosen profile:

```
$ /opt/ericsson/cmco/traceprofiles/bin/imstrace
<traceprofilename>
```

7. Give a user as search string, if necessary.

```
: 1000010@
```

```
-----
Collecting domains... done.
Verifying domains.... done.
Preparing session.... done.
Uploading session.... done.
```



```
-----  
Waiting for trace output.
```

8. Apply test cases for calls and traffic.
9. Use Ctrl+c to stop the script:

Note: This step does not stop the tracing, to end the started AppTrace, Step 10 must be performed.

10. End the apptrace/imstrace session:

```
$ /opt/ericsson/cmco/traceprofiles/bin/imstrace stop
```

These traces are output to Applog. See Section 3.3.1 MTAS Applog on page 8 for how to check the Applog.

3.5 Counters

The performance counters generated by the MTAS are another way to get useful information when troubleshooting a problem.

The performance counter files are generated in 3GPP compliant XML format and could be transferred outside the system for post-processing.

For more information about file format, refer to *Managed Object Model (MOM)*.

Alarms based on counter-thresholds can be defined to get an alarm when a counter exceeds a defined counter threshold. This is a way to get useful information when troubleshooting a problem. For more information about threshold-based alarms, refer to *Performance Management*.

In case alarms based on counter-threshold are raised, do a node health check, see Section 4.1 Perform Node Health Check on page 17.

3.6 Software Level Checks

This section describes how to check the software version.

3.6.1 Print the MTAS Software Version

To print the MTAS software version:

1. Log on to primary SC (SC-2 in the example), using telnet and the root privilege

```
telnet <address><port>  
SC-2 login: root  
Password:
```



2. Enter the password.

3.

```
SC-2:~ /opt/com/bin/cliss
>ManagedElement=1,MtasFunction=MtasFunction
(MtasFunction=MtasFunction) >show
MtasFunction=MtasFunction
  mtasFunctionAdministrativeState=UNLOCKED
  mtasFunctionVersion="CXP9023564 R2A05"
  MtasCharging=0
  MtasComDetails=0
  MtasCommonData=0
  MtasCsi=0
  MtasExtPowerSystem=0
  MtasGls=0
  MtasLicenses=0
  MtasMediaFramework=0
  MtasMrfc=0
  MtasPx=0
  MtasServices=0
  MtasSh=0
  MtasSip=0
  MtasSubsData=0
  MtasWebServices=0
  MtasWsNameDb=0
  MtasXdms=0
  MtasSystemConstant=0
```

4. Check the value of `mtasFunctionVersion`:

Example output:

```
mtasFunctionVersion="CXP9023564 R2A05"
```

3.7 Handle Log Files and Counter Files

This section describes how to handle log files and counter-files.

If an error occurs that requires that Ericsson support is contacted:

1. Copy the log files and counter files to a temporary directory when troubleshooting.
2. Transfer the files, using the Secure File Transfer Protocol (SFTP), to a local workstation or a PC.

Note: Deleting an active log file is not recommended.

3.8 Change Syslog Level

For information on how to change the syslog level to, for example, 7 (maximum output), refer to *Managed Object Model (MOM)*.



3.9 Restart

This section describes how to restart MTAS.

3.9.1 Do Small Restart

The small restart involves the restarting of all static processes, and termination of all dynamic processes. It keeps all data in the database and does not affect stable sessions. It is used to restart the MTAS SW without doing a cluster reboot.

To do a small restart:

1. In the COM/CLI:

```
>ManagedElement=1,MtasFunction=MtasFunction,mtasFunctionSmallRestart
```

2. A small restart is performed.

3.10 Recover H.248 Link

Note: This section is only needed if internal Media Resource Function Controller (MRFC) is used and the H.248 links are failing after the upgrade.

If the H.248 link does not come back after the upgrade, do a small restart. For more details on small restart, see Section 3.9.1 Do Small Restart on page 13.

If the H.248 link does not work after the small restart:

Do the following steps to deactivate the MRFP with the failing H.248 link:

1. From the COM/CLI, navigate to the `MtasMrfpNode` MO to be deactivated:

```
>configure
>ManagedElement=1,MtasFunction.applicationName=MtasFunction,MtasMediaFramework.mtasMediaFramework=0,MtasMrf.mtasMrf=0,MtasMpController.mtasMpController=0
```

2. Set the `mtasMrfpNodeAdministrativeState` attribute to `Shuttingdown`.
3. Commit
4. Wait until the network is free from traffic sessions, that is, typically more than five times the normal holding time for a session. Refer to call statistics for a suitable time.
5. Set the `mtasMrfpNodeAdministrativeState` attribute to `Locked`.
6. Perform a backup. For more information, refer to *Create Backup*.



Do the following steps to activate the MRFP again:

1. Navigate to the `MtasMrfpNode` MO to be activated in the COM/CLI:

```
>configure
>ManagedElement=1,MtasFunction.applicationName=MtasFunction,MtasMediaFramework.mtasMediaFramework=0,MtasMrf.mtasMrf=0,MtasMpController.mtasMpController=0
```

2. Set the `mtasMrfpNodeAdministrativeState` attribute to `Unlocked`.
3. Commit
4. Perform a backup. For more information, refer to *Create Backup*.

Refer to *MTAS Media Control Management Guide* for the details on activating and deactivating the MRFP.

If the MRFP traffic does not recover, from one of the SCs, SSH to each PL and execute the `reboot` command.

3.11 Sessions Hanging

If an MTAS node experiences problems of hanging sessions (which can potentially prevent a subscriber from establishing new sessions), the CM attribute `mtasFunctionMaxNumberOfSessionsAction` can be used to clear such hangings.

When an action is taken as a result of configuration, the session-related information including the event history, is printed to the Proc-logs.

When the number of sessions are equal to the CM attribute `mtasFunctionMaxNumberOfSessions`, the following setting decides the action to be taken:

- 0: reject new communication attempts.
- 1: gracefully terminate all sessions of the subscriber that exceed the maximum duration limit defined by the CM attribute `mtasFunctionMaxSessionDuration`.
- 2: forcibly terminate all system resources allocated by the subscriber, that is, Capsule Abortion.
- 3: reject new communication attempts. In addition to a possible reject, the MTAS also gracefully terminates sessions exceeding the maximum duration limit defined by `mtasFunctionMaxSessionDuration`, after each communication attempt to or from the subscriber.

Also, two more options exist for the clean-up mechanism:



- The presence of `MtasSystemConstantSC` ID 2 suppresses the continuous removal of non-stable sessions at application process serialization, hence all sessions are serialized without considering their state.
- The presence of `MtasSystemConstantSC` ID 3 suppresses the printout of session debug information to log when an action is taken as a consequence of the configuration of the CM attribute `mtasFunctionMaxNumberOfSessionsAction`.





4 Troubleshooting Procedure

This section describes the procedure for troubleshooting the MTAS. The procedure is basically to check the node health and alarms and to verify that the node is correctly configured.

The workflow is as follows:

- Performing a node health check to gather information about the state of the node
- Checking the alarms
- Checking the notifications
- Verifying the node configuration

Once the cause of the problem has been identified, refer to Section 5 on page 23 for information on how to recover from the problem. If the problem still cannot be solved, refer to Section 6 on page 29.

4.1 Perform Node Health Check

Information about the health check procedure can be found in *MTAS Health Check*.

4.2 Check Alarms

Alarms are accessible through the COM/CLI.

```
>ManagedElement=1, SystemFunctions=1, Fm=1
```

To handle an active alarm on the node:

1. Check the alarm log for any alarms.
2. If any alarms exist, follow the relevant Operating Instruction (OPI) and solve the problem that caused the alarm.

For more information regarding the MTAS alarms, refer to *MTAS Alarm List*.

For each of the CBA components alarms and notifications, there is an alarm OPI. For more information about the OPIs, refer to the relevant documents in the component library folder.

4.3 Verify Node Configuration

This section describes how to verify that the node is correctly configured.

To verify that the basic configuration of the node is correct:

1. Verify that the MTAS is activated.
2. Verify that the used services are enabled.

For more information, refer to the following documents:

- *MTAS Node Management Guide*
- *MTAS Service Management Guide*

3. Verify that all required licenses are installed.

For more information about the MTAS licenses, refer to *MTAS Licenses*.

4. Verify that the Diameter links are correctly configured and working.

For more information, refer to the following documents:

- *MTAS Charging Management Guide*
- *MTAS Media Control Management Guide*
- *MTAS Subscriber Data Management Guide*
- *MTAS XDMS Management Guide*

5. Verify that the Ericsson Media Resource Function Processor (MRFP) links are correctly configured and working.

6. Verify that the MTAS SIP interfaces are correctly configured and working.

For more information, refer to *MTAS SIP Management Guide*.

7. Verify the state of the VIP ports.

For more information, refer to *Virtual IP Address Management*.

8. Verify that the Number Normalization is correctly configured, refer to the following documents:

- *MTAS Number Normalization Management Guide*
- *Managed Object Model (MOM)*

9. If any of the items from step 1 to step 8 is faulty (incorrectly configured or in a faulty state), corrective actions must be taken. If the problem still cannot be solved, contact the next level of maintenance support.



4.4 Troubleshoot Scale in Failures

4.4.1 PL-3 or PL-4 Is Tried to Be Scaled In

The Payload nodes defined during maiden installation, like PL-3 and PL-4, cannot be scaled in. In that case the Scale In operation is rejected with the following response:

```
>ManagedElement=1, SystemFunctions=1, SysM=1, CrM=1, ComputerResourceRole=PL-4
```

```
(ComputeResourceRole=PL-4) >configure
```

```
(config-ComputeResourceRole=PL-4) >no provides
```

```
(config-ComputeResourceRole=PL-4) >up
```

```
(config-CrM=1) commit
```

```
ERROR: Transaction not committed due to validation errors
```

```
Transaction validation failed!
```

```
(config-CrM=1)
```

Recommended solution:

- Scale In a node from the traffic domain which was not defined during maiden installation.

4.4.2 Scale in Rejected by the Cluster Because of Insufficient Memory Capabilities of Target Sized Cluster

The cluster can reject the Scale in operation, see example output:

```
>ManagedElement=1, SystemFunctions=1, SysM=1, CrM=1, ComputerResourceRole=PL-5
```

```
(ComputeResourceRole=PL-5) >configure
```

```
(config-ComputeResourceRole=PL-5) >no provides
```

```
(config-ComputeResourceRole=PL-5) >up
```

```
(config-CrM=1) commit
```

```
ERROR: Transaction not committed due to validation errors
```

```
Transaction validation failed!
```

```
(config-CrM=1)
```



One possible reason for the rejection is that the cluster estimated the needed memory for the ongoing traffic and this requirement cannot be met if the Scale In operation is performed. Therefore to avoid traffic loss the cluster rejects the operation.

The procedure checking the memory consumption of the cluster is the same as described in *DBS, Memory Limit Reached*.

Recommended solutions:

- If multiple nodes were tried to be scaled in parallel, it is recommended to scale in the nodes one by one until the memory capabilities are still enough to handle the traffic.
- If even one node's scaling in is rejected the contraction of the system is not allowed to secure the needed capabilities.
- The memory consumption of the cluster can be decreased if the Subscriber Data is purged from the cluster, see *MTAS Subscriber Data Management Guide* for reference.

4.4.3 Unexpected Cluster Reboot during Scale In

If there is a cluster reboot during the Scale In operation, the cluster can result in an inconsistent state where the node is removed but some information remains about it. For example the Compute resource can still be seen:

```
(ComputeResourceRole=PL-5) >show
```

```
ComputeResourceRole=PL-5
```

```
adminState=LOCKED
```

```
instantiationState=UNINANTIATION_FAILED
```

```
operationalState=DISABLED
```

```
uses="ComputeResource=PL-5"
```

Note: This inconsistent state has no effect on the functional capabilities of the system. The Scale In operation was successful and the future auto scale out operations are also not prohibited in spite of the inconsistent state. Though this PL-5 entry cannot be scaled in again. That has affect on the system's capability as this ComputeResourceRole, PL-5, cannot be assigned any more, which means that the maximum cluster size cannot be reached any more.

Recommended solution for clearing the inconsistent information:

1. Roll back the system to the previous correct state by restoring the backup created before the Scale In operation, see *Restore Backup* for reference.



2. Perform the Scale In operation again

4.5 Troubleshoot License Manager

This section describes the procedure for enabling and disabling the License Manager related trace.

License Manager Functionality related events will be stored in log files, which is located in the following path of the cluster.

```
/storage/clear/lm-apr9010503/log/lm.SC-X.log
```

Where X is the number designation of the blade.

4.5.1 Enable the trace

To enable the trace do the following:

1. Log on to the Managed Element (ME) to access a Linux® shell:

```
ssh <user>@<hostname> -p 22
```

2. Edit /storage/system/config/lm-apr9010503/etc/lm_trace.cfg as follows:

```
Set the TraceState to 1 (a change from 0 to 1 enables tracing).
```

```
Set the TraceLevel to a log level from below list.
```

- 0: TRACE_LEVEL_EMERG
- 1: TRACE_LEVEL_ALERT
- 2: TRACE_LEVEL_CRIT
- 3: TRACE_LEVEL_ERR
- 4: TRACE_LEVEL_WARNING
- 5: TRACE_LEVEL_NOTICE
- 6: TRACE_LEVEL_INFO
- 7: TRACE_LEVEL_DEBUG

Note: It is important to disable the trace once needed information is collected.

4.5.2 Disable the trace

To disable the trace do the following:

1. Log on to the Managed Element (ME) to access a Linux® shell:



```
ssh <user>@<hostname> -p 22
```

2. Edit /storage/system/config/lm-apr9010503/etc/lm_trace.cfg as follows:

Set the TraceState to 0 (a change from 1 to 0 disable tracing).



5 Application Recovery Procedure

This section describes the application recovery procedure.

5.1 Diameter Interface

This section describes the Diameter interface trouble cases and how to solve them.

The following Diameter trouble cases are described:

- The Diameter interface is down
- Configuration fault
- Link inactivity
- IP Network failure
- System error

5.1.1 Diameter Interface Is Down

One or more of the Diameter interfaces are down.

5.1.1.1 Symptoms

Either of the following Diameter interfaces is down:

- Rf
- Ro
- Sh
- Dh

5.1.1.2 Locate and Confirm the Fault

The causes for the Diameter interface fault can be the following:

- Home Subscriber Server (HSS) is down
- Subscriber Location Function (SLF) is down
- Charging Server is down



- Communications Details server is down
- Network issues
- Format error of CER/CEA messages
- Coding error or missing Attribute-Value Pairs (AVP). In addition, if a vendor defined, mandatory AVP is received, but not defined on the receiving node, this error occurs.

To locate the Diameter interface fault:

1. Wait for an automatic reconnection.
2. If the connection is not established, disable the neighbor node and enable it again. For more information, refer to *Diameter Management*.
3. If a connection is still not established, consult the next level of maintenance support.

5.1.2 Link Inactivity

A connection to a Diameter peer is broken owing to link inactivity.

5.1.2.1 Symptoms

There is no response to the watchdog messages.

5.1.2.2 Locate and Confirm the Fault

In case of link inactivity:

1. Check if the other Diameter node is operational by contacting the system administrator of the neighbor node, and wait for automatic reconnection.
2. Disable the Diameter neighbor node temporarily and then enable it again. For more information, refer to *Diameter Management*.
3. If the connection is still not established, consult the next level of maintenance support.

5.1.3 IP Network Failure

A connection to a Diameter peer is broken owing to IP Network failure.

5.1.3.1 Symptom

There is an IP Network or socket failure, or a malformed message.



5.1.3.2 Locate and Confirm the Fault

If there is an IP Network failure:

1. Check the IP Network and wait for automatic reconnection.
2. Disable the Diameter neighbor node temporarily and then enable it again. For more information, refer to .
3. If the connection is still not established, consult the next level of maintenance support.

5.1.4 Check Diameter Link Status

A Diameter link is broken.

5.1.4.1 Symptom

The Diameter link has incorrect status in the COM/CLI.

5.1.4.2 Locate and Confirm the Fault

To check that all Diameter links are operational:

From the COM/CLI:

1. `>configure`
2. `(config)>ManagedElement=1,MtasFunction=MtasFunction,MtasSupportFunctions=0,DIA-CFG-Application=DIA,DIA-CFG-StackContainer=MTAS_SH,DIA-CFG-PeerNodeContainer.peerNodeContainerId=MTAS_SH`
3. Go through each `DIA-CFG-NeighbourNode=...\.MTAS_SH` and verify that the value of the attribute `linkStatus` is `Up` for each `DIA-CFG-Conn=MTAS_SH\...\23conn..`
4. `>..` (four times to get back to the config prompt)
5. `(config)>ManagedElement=1,MtasFunction=MtasFunction,MtasSupportFunctions=0,DIA-CFG-Application=DIA,DIA-CFG-StackContainer=MTAS_SH,DIA-CFG-PeerNodeContainer.peerNodeContainerId=MTAS`
6. Go through each `DIA-CFG-NeighbourNode=...\.MTAS` and verify that the value of the attribute `linkStatus` is `Up` for each `DIA-CFG-Conn=MTAS\...\23conn..`
7. `>..` (four times to get back to the config prompt)
8. `(config)>ManagedElement=1,MtasFunction=MtasFunction,MtasSupportFunctions=0,DIA-CFG-Application=DIA,DIA-CFG-S`



```
tackContainer=MTASXDMS,DIA-CFG-PeerNodeContainer.pee  
rNodeContainerId=MTASXDMS
```

9. Go through each `DIA-CFG-NeighbourNode=...\..MTASXDMS` and verify that the value of the attribute `linkStatus` is Up for each `DIA-CFG-Conn=MTASXDMS\...\23conn..`

If each one of the `linkStatus` is Up, the Diameter link is operational, as required.

If any of the `linkStatus` is not Up, some instance of the Diameter link is down and the next level of maintenance support must be consulted.

5.2 Handle Hung Carrier Select Synchronization

This section describes how to handle situations when one of the following Carriers Select synchronization CM parameters hangs in `TRUE` state:

- `CarSelDialedStringAnalysisTableSynchronization`
- `CarSelCarrierTableSynchronization`

Note: The procedure described in this section is not for daily use and must be performed with care.

For more Carrier Select synchronization information, refer to *class CarSel-Application*.

Resolving the hanging is performed through environment variable named `CARSEL_RESTART_SYNC`.

The `CARSEL_RESTART_SYNC` variable controls both the `CarSelDialedStringAnalysisTableSynchronization` and `CarSelCarrierTableSynchronization` parameters.

5.2.1 Indication of Hanged Synchronization

To have new configuration in operation, the CM parameter `CarSelDialedStringAnalysisTableSynchronization` or `CarSelCarrierTableSynchronization` is set to `TRUE`. The exact choice of parameter depends on which configuration is changed. The parameter is set to `TRUE` during the process of reading and caching of the data. When done, the parameter is set to `FALSE`. The whole process usually finishes in less than 10 minutes.

If however, the parameter is still set to `TRUE` after 60 minutes, it is an indication that the synchronization function has hung. One or more of the following procedures then must be performed.



5.2.2 Resynchronize

The first step to do when the synchronization hangs is to try synchronizing another time. Doing this is not possible with the CM parameters as they are already in `TRUE` state. Instead, set the environment variable `CARSEL_RESTART_SYNC` to “resync”:

To resynchronize:

1. Log on to OAM VIP of the target node, using SSH and the root privilege

```
ssh <root>@<OAM VIP Address>
Password:
```

2. Start the synchronization again:

```
$ immcfg -c LPMSvConfigAttribute -a lpmsvCfgAttrVal=resync lpmsvCfgAttr=CARSEL_RESTART_SYNC,lpmsv=LPMSvSite
```

3. Check if the synchronization is successful this time:

Monitor the same parameter as before, when it is set to `FALSE`, the synchronization is done. Then perform the procedure in Page 23. If the parameter is still set to `TRUE` after 60 minutes, the synchronization function has hung again. Perform the procedures in the following sections.

5.2.3 Collect Data about Hanging

To investigate the synchronization hanging, it is possible and important to perform a data dump and send it to MTAS for further analysis.

To collect data:

1. Dump the information.

```
immcfg -c LPMSvConfigAttribute -a lpmsvCfgAttrVal=dump lpmsvCfgAttr=CARSEL_RESTART_SYNC,lpmsv=LPMSvSite
```

2. Collect all console logs and send to MTAS.

Pack all console logs and include them in a report.

The console logs are located in the following directory:

```
/cluster/storage/no-backup/cdclsv/log/lpmsv
```

More information on Console logs, refer *MTAS Logs*



5.2.4 Stop Synchronization

When the synchronization fails, it is possible to force the CM parameters to FALSE.

To stop synchronization:

1. Force-stop the synchronization.

```
immcfg -c LPMSvConfigAttribute -a lpmsvCfgAttrVal=reset  
lpmsvCfgAttr=CARSEL_RESTART_SYNC,lpmsv=LPMSvSite
```




6 Trouble Report

Problems identified that cannot be solved by using this document must be reported to the next level of maintenance support.

Send a Trouble Report (TR), for internal Ericsson use, or a Customer Service Request (CSR) to the local Ericsson support organization. Include the data mandated by the Operating Instruction *Data Collection Guideline for MTAS*. Provide also a severity statement expressing how the customer and its subscriber are effected by the problem. Indicate if a temporary work-around exists.