

MTAS Health Check

MTAS

OPERATING INSTRUCTIONS

Copyright

© Ericsson AB 2016–2018. 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	Health Check Procedure	3
2.1	Execution of Automatic Health Check	3
2.2	Health Check Results	6
2.3	SLA Results	8
2.4	Health Check Verdict	8
3	Health Check Steps	11
3.1	AlarmsAndNotifications	11
3.2	AllMtasPortStatus	11
3.3	BackupList	11
3.4	ChargingBackupEvents	12
3.5	CoreMWStatus	12
3.6	CpuLoadOnPLs and CpuLoadOnSCs	12
3.7	DiameterPortsStatus	13
3.8	DiskUsageOnSCs	14
3.9	DrbdStatus	14
3.10	eVIP	14
3.11	MemoryUsageOnPLs and MemoryUsageOnSCs	15
3.12	Mmas	15
3.13	NETCONFConnection	16
3.14	NeLSConnectivity	16
3.15	NetworkConnectivity	16
3.16	NodeOutage	17
3.17	OngoingQueryPurge	17
3.18	OperationalState	17
3.19	SIPPortsStatus	18
3.20	SS7Connections	18
3.21	Sla	18
3.22	SecurityStatus	19
3.23	SoftwareInventory	19
3.24	SoftwareVersions	20



3.25	SystemEnvironmentVariables	20
3.26	SystemStatus	21
3.27	VirtualDicosProcessOutage	21
3.28	VmLogs	21
3.29	XdmsCaiLicence	22
3.30	XdmsInstance	22
3.31	XdmsRpm	22
3.32	XdmsTrafficApps	22
4	Health Check Profiles	25
4.1	HcMtasBasic	25
4.2	HcMtasFull	25
4.3	HcMtasPreUpgrade	26
4.4	HcMtasPostUpgrade	27
5	Problem Reporting	29



1 Introduction

This document describes how to perform the health check on the MTAS running in virtualized environment. The health check tasks described in Section 2 on page 3 are recommended to be performed before and after a system update or upgrade, a normal backup, or during the periodic maintenance.

1.1 Prerequisites

This section states the prerequisites for performing the health check procedure.

1.1.1 Documents

Before starting this procedure, ensure that the following information or documents are available:

- The release information for the MTAS software level that is intended to be run in the MTAS and MTAS RDP versions.

Note: The release information can, for example, be found in delivery reports, delivery specifications, delivery notes, release notes, or correction notes.

1.1.2 Knowledge

It is assumed that the user of this document is familiar with the Operation and Maintenance (O&M) area, in general. It is also assumed that the user is familiar with the concepts, terminology, and abbreviations within this area.

1.1.3 Tools

The following tool is required to check a summary of the health check:

- Any web browser supporting HTML 4.01.





2 Health Check Procedure

Health Check consists of a set of checks which verifies the status of the cluster, its fundamental functions, services, and external interfaces. These checks are called Health Check steps.

All steps are grouped into the following profiles:

- Basic
- Full
- PreUpgrade
- PostUpgrade

For a detailed description of the profiles, see Section 4 on page 25.

The Basic profile contains basic checks that determine decision of the MTAS node health status. The MTAS node can be considered healthy if all the checks are OK. By default, Health Check with Basic profile is performed periodically once per hour, but the periodicity is possible to change; see Section 2.1.2.2 Scheduled Periodic Health Check on page 6. In troubleshooting situations or when more information is desired, the checks can be performed manually, optionally with a broader profile.

When the execution of a profile is finished, a final verdict is produced by the Health Check. The result is written to the XML and HTML reports.

The result of the Health Check can be the following:

- OK (0) when the return of the steps is OK or INFO.
- VERIFY (2) when at least one of the steps return with VERIFY, and the others return with INFO, OK.
- FAIL (3) when at least one of the steps returns with FAIL and the others return with VERIFY, INFO, OK.
- ERROR (255) when at least one of the steps return with ERROR, and the others return with FAIL, VERIFY, INFO, OK.

2.1 Execution of Automatic Health Check

Health Check is integrated to the Crash Dump and Console Log Collection Service (CDCLS). Health Check can be performed by executing CDCLS packer objects. Refer to *LEM Error Dump and Log User Guide*.



It is possible through the Ericsson Command-Line Interface (ECLI) or directly from the System Controller.

2.1.1 Health Check Using ECLI

The Health Check profile packers are listed with the `cdclsv-list` command and executed with the `cdclsv-invoke` command. Execution status can be checked by `cdclsv-status` command.

Steps

1. Log on to the ECLI:

```
ssh <username>@<oam-vip> -p 830 -t -s cli
```

2. Check which profiles are available:

```
cdclsv-list
```

The following is an example output:

```
cdclsPk=DcMtasBasic,cdcls=CDCLSVSite
cdclsPk=DcMtasFull,cdcls=CDCLSVSite
cdclsPk=DcMtasLimited,cdcls=CDCLSVSite
cdclsPk=DcMtasSla,cdcls=CDCLSVSite
cdclsPk=HcMtasBasic,cdcls=CDCLSVSite
cdclsPk=HcMtasFull,cdcls=CDCLSVSite
cdclsPk=HcMtasPostUpgrade,cdcls=CDCLSVSite
cdclsPk=HcMtasPreUpgrade,cdcls=CDCLSVSite
```

3. Select one of the profiles listed in the previous step:

```
HcMtasBasic
HcMtasFull
HcMtasPostUpgrade
HcMtasPreUpgrade
```

4. Start health check with the selected profile:

```
cdclsv-invoke
```

```
cdclsPk=<profile>,cdcls=CDCLSVSite
```

For example, with the `HcMtasBasic` profile: `cdclsv-invoke`
`cdclsPk=HcMtasBasic,cdcls=CDCLSVSite`

5. Check the status of the packing:

```
cdclsv-status
```

```
cdclsPk=<profile>,cdcls=CDCLSVSite
```




Example with the `HcMtasBasic` profile:

```
cdclsv-status cdclsPk=HcMtasBasic,cdcls=CDCLSVSite
```

The results are found in:

```
/cluster/storage/no-backup/hc/
```

2.1.2 Health Check Using CDCLS

2.1.2.1 Manually Started Health Check

The Health Check profile packers are listed with the `cdclsv-list-packers` command and executed with the `cdclsv-pack` command. Execution status can be checked with the `cdclsv-pack-status` command.

Steps

1. Check which profiles are available:

```
cdclsv-list-packers | grep cdclsPk=HcMtas
```

The following result is shown:

```
cdclsPk=HcMtasFull,cdcls=CDCLSVSite
cdclsPk=HcMtasBasic,cdcls=CDCLSVSite
cdclsPk=HcMtasPostUpgrade,cdcls=CDCLSVSite
cdclsPk=HcMtasPreUpgrade,cdcls=CDCLSVSite
```

2. Select one of the profiles listed in the previous step:

```
HcMtasFull
HcMtasBasic
HcMtasMediumPriority
HcMtasPostUpgrade
HcMtasPreUpgrade
```

3. Start health check with the selected profile:

```
cdclsv-pack cdclsPk=<profile>,cdcls=CDCLSVSite
```

Example with the `HcMtasBasic` profile:

```
cdclsv-pack cdclsPk=HcMtasBasic,cdcls=CDCLSVSite
```

4. Check the status of the packing:

```
cdclsv-pack-status cdclsPk=<profile>,cdcls=CDCLSVSite
```

Example with the `HcMtasBasic` profile:

```
cdclsv-pack-status cdclsPk=HcMtasBasic,cdcls=CDCLSVSite
```



The results are found in:

```
/cluster/storage/no-backup/hc/
```

2.1.2.2 Scheduled Periodic Health Check

Health Check profiles are possible to schedule for periodic execution. By default, the Basic profile is executed in every hour, while other profiles are not scheduled for automatic execution.

The configured period can be read with the `cdclsv-get-pack-period` command, for example:

```
cdclsv-get-pack-period cdclsPk=HcMtasBasic,cdcls=CDCLSVSite
```

To set or change periodicity, use the `cdclsv-set-pack-period` command, for example:

```
cdclsv-set-pack-period cdclsPk=HcMtasBasic,cdcls=CDCLSVSite 3600
```

The value 0 means that the scheduled execution is switched off.

The scheduling periods are synchronized to entire hours. For example: if the period is set to 1200 (20min), Health Check is executed at the 0th, 20th, and 40th minutes of every hour.

2.2 Health Check Results

Health Check results are stored in directory `/storage/no-backup/hc`. Each Health Check run results in a separate package, a gzipped tar archive which contains the checkers status.

Furthermore, the Health Check report HTML files are copied into the directory `/storage/no-backup/hc_reports`.

2.2.1 Contents of the Health Check Result Package

The package contains the following items:

- `summary.html`: Contains general information about checkers status in the HTML format.
- `summary.xml`: Contains general information about checkers status in the XML format.
- `<Checker name>.log`: Contains data gathered by Data Collection. This log is not available if a particular Health Check step does not return data to the log, meaning that the result of the checker is OK. The `sla.log` is



an exception, since the log file is generated regardless of the SLA checker result.

- *<Checker name>*: Directory with data gathered by Data Collection step. The directory is not available if a particular Health Check step does not copy data or create specific structure. The Sla Directory is an exception, since this directory is generated regardless of the SLA checker result.

Data gathered by Data Collection is stored in result package only if any checker detects problems.

2.2.2 Housekeeping of the Results of Health Check

Housekeeping is required, as the results are collected cumulatively.

The housekeeping of directory `/storage/no-backup/hc` is configured in the ECLI. The configuration can be changed and checked using the ECLI, for example:

```
>dn ManagedElement=1,SystemFunctions=1,FileM=1,FileGroupPolicy=healthCheck
```

```
>show -v
```

The following is an example output:

```
fileGroupPolicyId="healthCheck"
fullFileGroupAction=DISCARD_OLDEST
maxFileGroupSize=1048576
maxNumberFiles=0
retentionTime=1440
userLabel=[]
```

The housekeeping of directory `/storage/no-backup/hc_reports` is configured in the ECLI. The configuration can be changed and checked using the ECLI, for example:

```
>dn ManagedElement=1,SystemFunctions=1,FileM=1,FileGroupPolicy=healthCheckReports
```

```
>show -v
```

The following is an example output:

```
fileGroupPolicyId="healthCheckReports"
fullFileGroupAction=DISCARD_OLDEST
maxFileGroupSize=10240
maxNumberFiles=24
retentionTime=1440
userLabel=[]
```



2.3 SLA Results

The result of Sla step is an exception, since it is generated and stored regardless of the Sla checker result.

The Sla result contains the following items:

- `Vm_KPI_<Date_BeginTime_EndTime>.log` Contains the following measurement for each VM:
 - Central Processing Unit (CPU) Total and CpuSteal
 - Total Memory, Used, and Free Memory
 - Free and Used Disk for System Controllers (SCs)
- `PerCore_KPI_<Date_BeginTime_EndTime>.log` Contains the following measurement for each CPU:
 - CPU Total and CpuSteal
- `Network_KPI_<Date_BeginTime_EndTime>.log` Contains the following measurement for transmit and receive side of each interface of PLs:
 - Throughput
 - Total number of packets
 - Dropped packets
 - Error packets
- `Sla_Verdict_Details.log`:
Contains detailed information for each VERDICT.

Note: The `Begin_time` and `End_Time` in the KPI logs, shows the time interval for the KPI data collection.

2.4 Health Check Verdict

The result from the checks is stored in summary files. The verdict is a way to inform the user about status of the individual checks. The definitions of the different verdicts are shown in Table 1.

Table 1 Health Check Verdicts

Verdict Sign	Verdict	Description
,	INFO	Information for the user, not checked by the script.
.	OK	Automatic checked passed.
?	VERIFY	Manual verification needed.



Verdict Sign	Verdict	Description
!	FAIL	Problem detected by automatic health check.
E	ERROR	The automatic health check is not possible to execute, that is, the input data is not available, script update needed or system broken.





3 Health Check Steps

3.1 AlarmsAndNotifications

This step checks if there are any unresolved alarms or notification. If a non-OK verdict is given, an AlarmsAndNotifications directory is packed into the result package, where log files, containing the details of unresolved alarms and notifications, can be found for manual examination.

Verdict

OK	No unresolved alarms or notifications found.
VERIFY	When unresolved notifications or alarms of warning or minor severity levels found.
FAIL	When unresolved alarms of major or critical severity levels found.

3.2 AllMtasPortStatus

This step verifies if the MTAS ports are open.

Verdict

OK	When all the checked ports are open and the corresponding server accepts incoming connections.
VERIFY	(never)
FAIL	When any of the checked ports are closed or the corresponding server does not accept incoming connections.

3.3 BackupList

This step checks if there is an active backup available to restore.

Verdict when step is executed from PreUpgrade profile:

OK	When there is one or more backups in the system, which can be restored.
VERIFY	(never)
FAIL	When a backup operation is ongoing, or a primary restore candidate for software and configuration is missing.



Verdict when step is executed from any other profile:

OK	When there is one or more backups in the system, which can be restored.
VERIFY	(never)
FAIL	A primary restore candidate for software and configuration is missing.

3.4 ChargingBackupEvents

This step verifies if there are buffered charging events.

Verdict when step is executed from PreUpgrade profile:

OK	There are no buffered charging events.
FAIL	There are buffered charging events.
ERROR	When the report file related to <code>mtascharging</code> Performance Measurement (PM) job is missing, or the PM job itself is not running.

Verdict when step is executed from any other profile:

OK	When there are no buffered charging events.
VERIFY	In any other cases.

3.5 CoreMWStatus

This step verifies if there are any AMF entities with questionable health.

Verdict

OK	If CoreMW is available and its status is UNLOCKED.
VERIFY	If CoreMW is available, but its status is LOCKED.
FAIL	If CoreMW is not available.

3.6 CpuLoadOnPLs and CpuLoadOnSCs

These steps verify the CPU load for each core of each node (PLs or SCs), and the average CPU load on each node (PLs or SCs).

The limits (**max** and **average**) for comparison depend on the profile this step has been called from.



For the PreUpgrade profile, both of the limits are set to 30%.

For any other profile, the limits for comparison are the values of the environment variables (in the following order):

- 1 `LOAD_REG_CPU_MAX_LIMIT` is the limit for the **max** load of each core.

`LOAD_REG_CPU_AVG_LIMIT` is the limit for the **average** load of the cores of a CPU.

If the variables are defined.

- 2 If `LOAD_REG_CPU_MAX_LIMIT` or `LOAD_REG_CPU_AVG_LIMIT` are not defined, `LOAD_REG_LIMIT` is used, if defined.
- 3 If `LOAD_REG_LIMIT` is not defined, both limits (**max** and **average**) are set to 85%.

Verdict when step is executed from PreUpgrade profile:

OK	When CPU load is less than the limit (30%) for every node and every core.
FAIL	When CPU load of any core, or the average CPU load of any node, is higher than the limit (30%).

Verdict when step is executed from any other profile:

OK	When CPU load of any core is less than the max limit by at least 10%, and the CPU load of any node is less than the average limit by at least 10%.
VERIFY	When CPU load of any core is closer to the max limit than 10%, or the CPU load of any node is closer to the average limit than 10%.
FAIL	When CPU load of any core is higher than the max limit, or the average CPU load of any node is higher than the average limit.

3.7 DiameterPortsStatus

This step verifies Diameter ports status. Data about diameter port configuration is gathered from COM management objects.

Verdict



OK	If Diameter stack is configured and at least one link is in ESTABLISHED state.
VERIFY	(never)
FAIL	If Diameter stack is not configured or the links are not in ESTABLISHED state.

3.8 DiskUsageOnSCs

This step verifies the level of available space on SCs disks.

The verdict depends on the profile this step has been called from.

Verdict when step is executed from PreUpgrade profile:

OK	When available space is more than 15%.
FAIL	When available space is less than 15%.

Verdict when step is executed from any other profile:

OK	If available space is more than 25%.
VERIFY	If available space is less than 25%.
FAIL	If available space is less than 15%.

3.9 DrbdStatus

This step verifies whether the shared block device of the cluster is functioning correctly. Connection state, disk state, and out-of-sync blocks are verified.

Verdict

OK	If all the verifications passed without errors
VERIFY	(never)
FAIL	If DRBD is in a disconnected or inconsistent state.

3.10 eVIP

This step verifies status of eVIP on all active ALBs.

Verdict



OK	If none of the eVIP agents are in INACTIVE or DOWN or REGISTERED or PENDING or INI state.
VERIFY	(never)
FAIL	If any of the eVIP agents are in INACTIVE or DOWN or REGISTERED or PENDING or INI state.

3.11 MemoryUsageOnPLs and MemoryUsageOnSCs

This step checks the memory use on the nodes (PLs or SCs).

The limit for comparison is the value of the environment variable (in the following order):

- 1 `LOAD_REG_MEMORY_LIMIT`, if defined.
- 2 If `LOAD_REG_MEMORY_LIMIT` is not defined, the limit is the value of `LOAD_REG_LIMIT`, if defined.
- 3 If none of these variables are defined, the limit is set to 85% of the available memory.

The verdict depends on the profile this step has been called from.

Verdict when step is executed from PreUpgrade profile:

OK	When memory use is less than the limit.
FAIL	When memory use is higher than the limit.

Verdict when step is executed from any other profile:

OK	If memory use is less than the limit by at least 10%.
VERIFY	If memory use is closer to the limit than 10%.
FAIL	If memory use is higher than the limit.

3.12 Mmas

This step verifies whether MMAS traffic instances are operational on every payload node.

Verdict



OK	If traffic instance is running on each PL.
VERIFY	(never)
FAIL	If traffic instance is not running on any of the PLs.

3.13 NETCONFConnection

This step verifies if NETCONF is configured on only one controller.

Verdict

OK	If NETCONF is correctly configured on only one SC node.
VERIFY	(never)
FAIL	If NETCONF is configured on more than one node. If NETCONF is not configured at all or configuration is faulty.

3.14 NeLSConnectivity

This step verifies the connectivity between the MTAS and the NeLS server.

Verdict

OK	If the NeLS server is configured and the connection between MTAS and NeLS server is operational.
VERIFY	If the NeLS server is configured and the connection between the MTAS and NeLS server is not established until for 24 hours.
FAIL	If the NeLS server is not configured. Or If the NeLS server is configured and the connection between MTAS and NeLS server is not established for more than 24 hours.

3.15 NetworkConnectivity

This step verifies the connectivity between each SC/PL node.

Verdict



OK	If connectivity between SCs/PLs is appropriate.
VERIFY	(never)
FAIL	If packet loss was detected while transferring test data between any two SCs/PLs.

3.16 NodeOutage

This step verifies SCs/PLs state and checks for recovery events in the last 24 hours of ISP logs.

Verdict

OK	If all the SCs and PLs are started and last 24 hours of ISP log does not indicate the occurrence of automatic recovery events.
VERIFY	If automatic recovery events have occurred in the last 24 hours.
FAIL	If any of the nodes are not in started state.

3.17 OngoingQueryPurge

This step verifies that a QueryPurge operation is ongoing.

The verdict depends on the profile this step has been called from.

Verdict when the step is executed from the PreUpgrade profile:

OK	When NO ongoing query or purge operation is running.
FAIL	When there is an ongoing query or purge operation.

Verdict when the step is executed from any other profile:

OK	When NO ongoing query or purge operation is running.
VERIFY	When there is an ongoing query or purge operation.

3.18 OperationalState

This step checks MTAS operational state using COM interfaces.

Verdict



OK	If <code>mtasFunctionAdministrativeState</code> is in UNLOCKED state.
VERIFY	(never)
FAIL	If <code>mtasFunctionAdministrativeState</code> is in LOCKED state.

3.19 SIPPortsStatus

This step verifies if SIP ports are open.

Verdict

OK	If every SIP port is open on all the SCs.
VERIFY	(never)
FAIL	If any of the SIP ports are closed on any of the SCs.

3.20 SS7Connections

This step verifies SS7 stack status.

Verdict

OK	If there is an activated SS7 connection.
VERIFY	If SS7 stack is configured, but there is no active SS7 connection.
FAIL	If there is no status information found for SS7 stack.
INFO	If SS7 stack is not configured/activated.

3.21 Sla

This step verifies the status of Service Level Agreement (SLA) and records the Key Performance Indicator (KPI) for the Virtual Machine (VM), Core and Network Interface under the Sla Directory for the last hour.

Verdict



OK	<p>The Verdict is OK when all the following conditions are fulfilled:</p> <ul style="list-style-type: none"> • CpuSteal \leq 1% for each VM and each Core of VM • Package Loss \leq 0.1% for all the Interfaces of VM • No VM Outage is detected.
VERIFY	<p>The Verdict is VERIFY when any of the following conditions are fulfilled:</p> <ul style="list-style-type: none"> • Any VM outage is detected. • If any VM has left the cluster and is not joined.
FAIL	<p>The Verdict is FAIL when any of the following conditions are fulfilled:</p> <ul style="list-style-type: none"> • CpuSteal $>$ 1% for any VM or any Core of VM • Package Loss $>$ 0.1% for any Interface of VM

For more details on the SLA Results, refer to Section 2.3 SLA Results on page 8.

For information on how to troubleshoot SLA, refer to Section 4.6 in *MTAS Troubleshooting Guideline*.

3.22 SecurityStatus

This step verifies if Core MW security package is installed on the SC/PL nodes.

Verdict

OK	If Core MW security package is installed on each SC/PL node in the system.
VERIFY	(never)
FAIL	If Core MW security package is not installed on any of the SC/PL nodes in the system.

3.23 SoftwareInventory

This step collects a list of RPM/SDP files available on the SC/PL nodes.

Verdict



OK	(never)
VERIFY	(never)
FAIL	(never)
INFO	Returns the list of Load Modules and RedHat Packages which are existing on the system.

3.24 SoftwareVersions

This step collects the software components installed on the cluster and checks whether there are “not used” ones among them.

It also compares the installed non-MTAS components to the expected ones included in the software package.

The verdict depends on the profile this step has been called from.

Verdict when the step is executed from PreUpgrade profile:

OK	When all the installed components are “used” and the versions of non-MTAS components match the expected ones included in the SW package.
VERIFY	When “not used” components are found on the cluster.
FAIL	When a difference is detected between the installed non-MTAS components and the expected ones included in the SW package.

Verdict when the step is executed from any other profile:

OK	When all the installed components are “used” and the versions of non-MTAS components match the expected ones included in the SW package.
VERIFY	When “not used” components are found on the cluster or when a difference is detected between the installed non-MTAS components and the expected ones included in the SW package.

3.25 SystemEnvironmentVariables

This step checks whether the vDicos environment variables are set correspondingly to the reference values.

Verdict



OK	If every environment variable equals to the reference value or, where applicable, it is in the reference range.
VERIFY	If any of the environment variables equals to the warning level reference value or, where it is in a range which is acceptable with warning. Detailed information can be found in the report file.
FAIL	If any of the environment variables equals to some unacceptable value or, where applicable, is out of the acceptable range. Detailed information can be found in the report file.

3.26 SystemStatus

This step verifies the system services status. Data is gathered by cmw-status.

Verdict

OK	If cmw-status reports OK for every service.
VERIFY	(never)
FAIL	If cmw-status reports NOK for any service.

3.27 VirtualDicosProcessOutage

This step checks status of vDicos Virtual Machines.

Verdict

OK	If every vDicos VM is operational.
VERIFY	(never)
FAIL	If any of the vDicos VMs are in a faulty status.

3.28 VmLogs

This step inspects vDicos Virtual Machine Logs if severe error messages logged in the last 24 hours.

Verdict

OK	If log inspection is OK.
VERIFY	If number of error messages shows potential problem.
FAIL	(never)



3.29 XdmsCaiLicence

This step checks whether the XDMS server certificate is valid.

Verdict

OK	If SSL certificate exists and is not expired.
VERIFY	(never)
FAIL	If SSL certificate does not exist or expired.

3.30 XdmsInstance

This step verifies if the XDMS instance exists in MMAS and if its status is OK.

Verdict

OK	If status is OK.
VERIFY	(never)
FAIL	If XDMS instance does not exist or it is in a faulty status.

3.31 XdmsRpm

This step verifies if every necessary XDMS-related package is installed on the system.

Verdict

OK	If every necessary XDMS-related package is installed on the system.
VERIFY	(never)
FAIL	If any of the necessary XDMS-related packages are absent.

3.32 XdmsTrafficApps

This step verifies traffic instance logs from the MMAS server.

Verdict



OK	If traffic instance exists on each payload node and no severe messages are shown in the instance logs.
VERIFY	If warning or minor level messages are found in the instance logs.
FAIL	If MMAS traffic instance is not found on one or more PLs. If error, critical or major level messages are found in the instance logs.





4 Health Check Profiles

This section describes health check profiles content. All health checks are grouped according to importance: Mandatory, Medium, and Low priority.

4.1 HcMtasBasic

This profile contains checkers only for the most crucial parts of the system. Health Check using this profile is automatically performed every hour by default.

HcMtasBasic profile includes the following steps:

- AlarmsAndNotifications
- CoreMWStatus
- DrbdStatus
- eVIP
- Mmas
- NETCONFConnection
- NeLSConnectivity
- NetworkConnectivity
- NodeOutage
- OperationalState
- SS7Connections
- Sla
- SystemStatus
- VirtualDicosProcessOutage
- XdmsInstance

4.2 HcMtasFull

This profile contains every checker available. By using this profile, a comprehensive set of information is constructed about system health.



Full profile contains all steps included in the Basic profile and the following steps:

- AllMtasPortsStatus
- BackupList
- ChargingBackupEvents
- CpuLoadOnPLs and CpuLoadOnSCs
- DiameterPortsStatus
- DiskUsageOnSCs
- MemoryUsageOnPLs and MemoryUsageOnSCs
- OngoingQueryPurge
- SIPPortsStatus
- SecurityStatus
- SoftwareInventory
- SoftwareVersions
- SystemEnvironmentVariables
- VmLogs
- XdmsCaiLicence
- XdmsRpm
- XdmsTrafficApps

Running Health Check using this profile can cause CPU load peaks and increase of memory use on primary SC.

4.3 HcMtasPreUpgrade

This profile is intended to be used before upgrade execution.

The PreUpgrade profile contains all steps from the Full profile. The following steps produce different verdicts when they are called from this profile:

- BackupList
- ChargingBackupEvents
- OngoingQueryPurge



- SoftwareVersions

4.4 HcMtasPostUpgrade

This profile is intended to be used after upgrade execution.

The PostUpgrade profile contains all steps from the Full profile. The following step produces different verdicts when it is called from this profile:

- SoftwareVersions





5 Problem Reporting

For any abnormal situation, refer to *MTAS Troubleshooting Guideline*.

If the problem still exists, the user can report it to the next level of support.

It is also important to collect the related data. For more information, refer to *Data Collection Guideline for MTAS*.