

MTAS Health Check

MTAS

OPERATING INSTRUCTIONS

Copyright

© Ericsson AB 2016–2019. 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 Health Check	3
2.2	Health Check Results	7
2.3	SLA Results	11
2.4	Health Check Verdict	11
3	Checks/Steps	13
3.1	AlarmsAndNotifications	13
3.2	AllMtasPortsStatus	13
3.3	BackupList	13
3.4	ChargingBackupEvents	14
3.5	CmData	14
3.6	CoreMWStatus	15
3.7	CpuLoadOnPLs and CpuLoadOnSCs	15
3.8	DiameterPortsStatus	16
3.9	DiskUsageOnSCs	16
3.10	DrbdStatus	16
3.11	eVIP	17
3.12	MemoryUsageOnPLs and MemoryUsageOnSCs	17
3.13	Mmas	18
3.14	NETCONFConnection	18
3.15	NeLSConnectivity	18
3.16	NetworkConnectivity	18
3.17	NodeOutage	19
3.18	OngoingQueryPurge	19
3.19	OperationalState	20
3.20	SIPPortsStatus	20
3.21	SS7Connections	20
3.22	Sla	20
3.23	SecurityStatus	21
3.24	SoftwareVersions	21



3.25	SystemEnvironmentVariables	22
3.26	SystemStatus	22
3.27	TcpPortUsage	23
3.28	VirtualDicosProcessOutage	23
3.29	VmLogs	23
3.30	XdmsCaiLicence	23
3.31	XdmsInstance	24
3.32	XdmsRpm	24
3.33	XdmsTrafficApps	24
4	Health Check Profiles	25
4.1	Basic Type / HcMtasBasic Profile	25
4.2	Full Type / HcMtasFull Profile	25
4.3	Preupgrade Type / HcMtasPreUpgrade Profile	26
4.4	Postupgrade Type / HcMtasPostUpgrade Profile	27
5	Problem Reporting	29



1 Introduction

This document describes how to perform a 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, installation, or during 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

A health check consists of a set of checks, which verifies the status of the MTAS Virtual Network Function (VNF), its fundamental functions, services, and external interfaces. These checks are called “checks” or “steps”.

All checks/steps are grouped into “types”. Each type contains a predefined set of checks/steps. The types are also named “profiles”.

Note: The naming convention of types/profiles is slightly different.

- basic type = Basic profile
- preupgrade type = PreUpgrade profile
- postupgrade type = PostUpgrade profile
- full type = Full profile

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

The basic type contains basic checks that determine the decision of the MTAS VNF health status. The MTAS VNF can be considered healthy if all checks are OK.

By default, a health check with basic type is performed periodically once per hour, but the periodicity is possible to change, see Section 2.1.1 Health Check Using vMTASHealthCheck Script on page 3 and Section 2.1.3.2 Scheduled Periodic Health Check on page 7. In troubleshooting situations, or when more information is needed, the checks can be performed manually, optionally with a broader type.

When the execution of a type is finished, a final verdict is produced by the health check. The result is written to the XML and HTML reports.

2.1 Execution of Health Check

There are several ways to start a health check:

- Using the vMTASHealthCheck script, see Section 2.1.1 Health Check Using vMTASHealthCheck Script on page 3.
- Using the Ericsson Command-Line Interface (ECLI), see Section 2.1.2 Health Check Using ECLI on page 5
- Using the Crash Dump and Console Log Collection Service (CDCLS), see Section 2.1.3 Health Check Using CDCLS on page 6.

2.1.1 Health Check Using vMTASHealthCheck Script

Steps



1. Log on to a System Controller, for example:

```
ssh <username>@<oam-mip>
```

2. Start the health check script:

```
/opt/mtas/hc/scripts/vMTASHealthCheck
```

vMTASHealthCheck can execute the selected type, or the selected checks.

The tool is enabled to set periodic execution for basic and for full types.

Without any parameters, a basic-type health check is executed. Additional parameters can be used:

Parameters	Comment
-h, --help	
-v, --verbose	Additional logs are added to the HC report.
-t TYPE, -type TYPE	
-sp PERIOD -schedule_period PERIOD	Periodic execution can be configured by setting the time of period for the given type (basic or full). Expected values (hour): <ul style="list-style-type: none">• 1• 2• 3• 4• 6• 8• 12
-sg -schedule_get	Parameter to GET the time of periods for scheduled jobs (basic and full types) with IDs.
-ch CHECK [CHECK,...], -check CHECK [CHECK,...]	Lists the name of the required checks. (See the help for the exact names.)
-sd [ID_1,ID_2,...] / ALL -schedule_delete [ID_1,ID_2,...] / ALL	Delete scheduled jobs by ID(s).

Examples:



To display the help:

```
/opt/mtas/hc/scripts/vMTASHealthCheck -h
```

To start a health check with checks listed in the full type:

```
/opt/mtas/hc/scripts/vMTASHealthCheck -t full
```

To set periodic execution of basic type, period is 1 hour:

```
/opt/mtas/hc/scripts/vMTASHealthCheck -t basic -sp 1
```

To get the scheduled jobs:

```
/opt/mtas/hc/scripts/vMTASHealthCheck -sg
```

To start a health check with the checks AlarmsAndNotifications and BackupList:

```
/opt/mtas/hc/scripts/vMTASHealthCheck -ch AlarmsAndNotifications,BackupList
```

2.1.2 Health Check Using ECLI

Health check can be executed through the Ericsson Command-Line Interface (ECLI) or directly from the System Controller.

The health check profiles 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:
2. Start the health check with one of the following profiles:

```
HcMtasBasic
HcMtasFull
HcMtasPostUpgrade
HcMtasPreUpgrade
```

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

For example, with the HcMtasBasic profile:

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

To list all available health check and Data Collection profiles, use the command `cdclsv-list`.



3. Check the progress of the health check:

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

Example with the HcMtasBasic profile:

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

Check the progress of the health check periodically, while waiting for the process to complete. When the result is “Idle”, the process is completed.

2.1.3 Health Check Using CDCLS

2.1.3.1 Manually Started Health Check

The health check profiles 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. Log on to a System Controller, for example:

```
ssh <username>@<oam-mip>
```

2. Start the health check with one of the following profiles:

```
HcMtasFull
HcMtasBasic
HcMtasPostUpgrade
HcMtasPreUpgrade
```

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

Example with the HcMtasBasic profile:

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

To list all available health check profiles, use the command `cdclsv-list-packers | grep cdclsPk=HcMtas`.

3. Check the progress of the health check:

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

Example with the HcMtasBasic profile:

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

Check the progress of the health check periodically, while waiting for the process to complete. When the result is “Idle”, the process is completed.



2.1.3.2 Scheduled Periodic Health Check

Health check profiles can be scheduled for periodic execution. By default, the Basic profile is executed every hour on the hour, while other profiles are not scheduled for automatic execution.

Steps

1. Set or change periodicity, use the `cdclsv-set-pack-period` command. To set it, for example, to 6 hours:

```
cdclsv-set-pack-period cdclsPk=HcMtasBasic,cdcls=CDCLSVSite 21600
```

The default value is 3600 and means that the scheduled execution is executed every hour on the hour.

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. Read the configured period with the `cdclsv-get-pack-period` command, for example:

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

2.2 Health Check Results

Health check results are stored in the 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 `/var/filem/nbi_root/healthcheck/reports` which is a symlink to the directory `/storage/no-backup/hc_reports`.

2.2.1 Contents of Health Check Result Package

The package contains the following items:

- An XML and HTML report file, where the naming convention is the following:

```
vMTAS_HC_<vnfName>_<startTtime>_<typeOfCheck>
```

For example:

```
vMTAS_HC_MTAS_2019-03-11T19:09:07+0200_preUpgrade.xml
```

```
vMTAS_HC_MTAS_2019-03-11T19:09:07+0200_preUpgrade.html
```



The files contain detailed information about the started type/profile, and about the executed checks/steps. Every check result is completed with a reason and recommended action.

If verbose mode is switched on, the detailed logs are also available.

The derived result of the health check can be as follows:

- **HEALTHY**: when the return of the checks is OK or INFO.
- **NOT_HEALTHY**: when at least one of the checks returns with FAIL or VERIFY or ERROR.

— `<check_name>.log`:

Contains data gathered by the data collection check/step and contains the verdict-information from the health check.

— `<check_name>`:

Directory with data gathered by the data collection check/step. This directory is created only for the checks/steps, which are gathering large amounts of logs. The directory is available when the result of the check is not OK or verbose mode was switched on.

The SLA directory is an exception, since this directory is generated regardless of the SLA checker result or verbose mode.

— `summary.html` and `summary.xml` (old-style report):

This file contains a short list with the checks/steps and their results.

The derived result of the health check can be as follows:

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

2.2.2 Contents of `/storage/no-backup/hc_reports`

Every execution of a health check produces the following two HTML files:

— `vMTAS_HC_[profile-name]_[date]_[time].html`, which is a copy of `vMTASHealthCheckReport.html` from the result package (see Section 2.2.1 Contents of Health Check Result Package on page 7).



- MTAS_HC_[profile-name]_[date]_[time].html: (old-style report), which is a copy of summary.html from the result package (see Section 2.2.1 Contents of Health Check Result Package on page 7).

2.2.3 Housekeeping of the Results of Health Check

Housekeeping is required, as the results are collected cumulatively.

The housekeeping of directory /storage/no-backup/hc and /storage/no-backup/hc_reports is configured in the ECLI.

The configuration can be changed and checked using the ECLI.

Steps

1. Show the current configuration for /storage/no-backup/hc:

- a. Log on to the ECLI:

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

- b. Navigate to the healthCheck FileGroupPolicy MO, for example:

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

- c. Show the current configuration:

```
(FileGroupPolicy=healthCheck)>show -v
```

The following is an output example with the default values:

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

The attribute values are interpreted as follows:

- maxFileGroupSize: The unit is kilobyte. 0 indicates that no limit is set.
- maxNumberFiles: 0 indicates that no limit is set. There are also two small system files in the directory counted by the housekeeper. Thus, when the value is set to 4, only two health check results are available.
- retentionTime: The unit is minutes. 0 indicates that the files are kept forever.



2. Update the configuration for /storage/no-backup/hc:

- a. Log on to the ECLI:

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

- b. Navigate to the healthCheck FileGroupPolicy MO, for example:

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

- c. Enter configure mode:

```
(FileGroupPolicy=healthCheck)>configure
```

- d. Set the new value, for example:

```
(config-FileGroupPolicy=healthCheck)>maxNumberFiles=100
```

Note: Be careful when changing these values as disk space is limited.

There are also two small system files in the directory counted by the housekeeper. Thus, when the value of maxNumberFiles is set to 100, only 98 health check results are available.

- e. Commit the changes:

```
(config-FileGroupPolicy=healthCheck)>commit
```

- f. Verify the new values:

```
(FileGroupPolicy=healthCheck)>show -v
```

The following is an example output:

```
fileGroupPolicyId="healthCheck"
fullFileGroupAction=DISCARD_OLDEST
maxFileGroupSize=1048576
maxNumberFiles=100
retentionTime=1440
userLabel=[]
```

3. To show and update the current configuration for /storage/no-backup/hc_reports, use the same method as in Step 1 and Step 2, but change the MO to:

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



2.3 SLA Results

The result of the 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 the report 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	Checked passed.
?	VERIFY	Manual verification needed.



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



3 Checks/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	Unresolved notifications or alarms of warning or minor severity levels found.
FAIL	Unresolved alarms of major or critical severity levels found.

3.2 AllMtasPortsStatus

This step verifies if the MTAS TCP ports are open.

The MTAS TCP ports are:

- mtasXdmsXCAPPort: 8090
- mtasXdmsCCMPPort: 8096
- mtasSoapPort: 9080
- mtasXdmsCai3GSecurePort: 8443
- mtasXdmsCai3GPort: 8095

Verdict

OK	All the checked ports are open and the corresponding server accepts incoming connections.
FAIL	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 a backup available to restore.

Verdict when step is executed as part of the PreUpgrade profile:



OK	There are one or more backups, which can be restored, available in the system.
FAIL	There is a backup operation ongoing, or there is no backup to restore available in the system.

Verdict when step is executed as part of any profile, except the PreUpgrade profile:

OK	There are one or more backups, which can be restored, available in the system.
FAIL	There is no backup to restore available in the system.

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	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	There are no buffered charging events.
VERIFY	In any other cases.

3.5 CmData

This step verifies the changes of application configuration attributes (CM data) during the upgrade procedure. The step is executed from PostUpgrade profile.

Verdict

OK	No MTAS CM attributes have been deleted during the upgrade, and the values of the CM attributes have not been changed; except <code>mtasFunctionVersion</code> , MTAS release, <code>mtasFunctionAdministrativeState</code> , System Constants. (The values of these four attributes can be changed.)
VERIFY	Value of any other CM attribute has been changed, or any attributes has been deleted.



Note: The step produces the verdict only if the upgrade FROM state is 1.12 or later.

3.6 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.

3.7 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.

3.8 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.
FAIL	If Diameter stack is not configured or the links are not in ESTABLISHED state.

3.9 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%.

3.10 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
FAIL	If DRBD is in a disconnected or inconsistent state.

3.11

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.
FAIL	If any of the eVIP agents are in INACTIVE or DOWN or REGISTERED or PENDING or INI state.

3.12

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.



3.13 Mmas

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

Verdict

OK	If traffic instance is running on each PL.
FAIL	If traffic instance is not running on any of the PLs.

3.14 NETCONFConnection

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

Verdict

OK	If NETCONF is correctly configured on only one SC node.
FAIL	If NETCONF is configured on more than one node. If NETCONF is not configured at all or configuration is faulty.

3.15 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.

3.16 NetworkConnectivity

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

Verdict



OK	If connectivity between SCs/PLs is appropriate.
FAIL	If packet loss was detected while transferring test data between any two SCs/PLs.

3.17 NodeOutage

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

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

Verdict when the step is executed from PreUpgrade or PostUpgrade profile:

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

Verdict when the step is executed from any other profile:

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

3.18 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.



3.19 OperationalState

This step checks MTAS operational state using COM interfaces.

Verdict

OK	If <code>mtasFunctionAdministrativeState</code> is in UNLOCKED state.
FAIL	If <code>mtasFunctionAdministrativeState</code> is in LOCKED state.

3.20 SIPPortsStatus

This step verifies if SIP ports are open.

Verdict

OK	If every SIP port is open on all the PLs.
FAIL	If any of the SIP ports are closed on any of the PLs.

3.21 SS7Connections

If SS7 is not configured on the node, then this step is omitted from the HC report.

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.

3.22 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"> • $3\% > \text{CpuSteal} > 1\%$ for any VM or any Core of VM • 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 $> 3\%$ 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 11.

For information on how to troubleshoot SLA, refer to Section 4.6 in [MTAS Troubleshooting Guideline](#).

3.23 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.
FAIL	If Core MW security package is not installed on any of the SC/PL nodes in 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.
FAIL	If cmw-status reports NOK for any service.

3.27 TcpPortUsage

This step checks the TCP ports of MTAS and verifies whether there are enough ephemeral ports available.

Verdict

OK	There are enough ephemeral TCP ports available.
VERIFY	Warning-limit (over 90%) of ephemeral TCP port connections is exceeded.
FAIL	No more ephemeral TCP port connections are available.

3.28 VirtualDicosProcessOutage

This step checks status of vDicos Virtual Machines.

Verdict

OK	If every vDicos VM is operational.
FAIL	If any of the vDicos VMs are in a faulty status.

3.29 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.

3.30 XdmsCaiLicence

This step checks whether the XDMS server certificate is valid.

Verdict



OK	If SSL certificate exists and is not expired.
FAIL	If SSL certificate does not exist or expired.

3.31 XdmsInstance

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

Verdict

OK	If status is OK.
FAIL	If XDMS instance does not exist or it is in a faulty status.

3.32 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.
FAIL	If any of the necessary XDMS-related packages are absent.

3.33 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 types/profiles content. All checks/steps are grouped according to importance.

4.1 Basic Type / HcMtasBasic Profile

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

HcMtasBasic profile includes the following checks/steps:

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

4.2 Full Type / HcMtasFull Profile

This type/profile contains every checker available.



HcMtasFull 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
- SoftwareVersions
- SystemEnvironmentVariables
- VmLogs
- XdmsCaiLicence
- XdmsRpm
- XdmsTrafficApps

Running a health check using this profile can cause CPU load peaks and increase of memory use on the primary SC.

4.3 Preupgrade Type / HcMtasPreUpgrade Profile

This type/profile is intended to be used before upgrade execution.

The HcMtasPreUpgrade profile contains all steps from the full type/Full profile. The following checks/steps produce different verdicts when they are called from this type/profile:

- BackupList
- ChargingBackupEvents
- CpuLoadOnPLs and CpuLoadOnSCs
- DiskUsageOnSCs



- MemoryUsageOnPLs and MemoryUsageOnSCs
- NodeOutage
- OngoingQueryPurge
- SoftwareVersions

The following check/step is called from this type/profile but it does not produce any verdict:

- CmData

4.4 Postupgrade Type / HcMtasPostUpgrade Profile

This type/profile is intended to be used after upgrade execution.

The HcMtasPostUpgrade profile contains all steps from the full type/Full profile. The following check/step produces different verdicts when it is called from this profile:

- NodeOutage





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](#).