

MTAS Health Check

MTAS

OPERATING INSTRUCTIONS

Copyright

© Ericsson AB 2016, 2017. All rights reserved. No part of this document may be reproduced in any form without the written permission of the copyright owner.

Disclaimer

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

Trademark List

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	Packers List	3
2.2	Manual Health Check	3
2.3	Health Check Results	3
2.4	SLA Results	4
2.5	Health Check Verdict	5
3	Health Check Steps	7
3.1	AlarmsAndNotifications	7
3.2	AllMtasPortStatus	7
3.3	BackupList	7
3.4	CoreMWStatus	8
3.5	CpuLoadOnPLs	8
3.6	CpuLoadOnSCs	8
3.7	DiameterPortsStatus	8
3.8	DiskUsageOnSCs	9
3.9	DrbdStatus	9
3.10	eVIP	9
3.11	MemoryUsageOnPLs	9
3.12	MemoryUsageOnSCs	10
3.13	Mmas	10
3.14	NETCONFConnection	10
3.15	NetworkConnectivity	11
3.16	NodeOutage	11
3.17	OperationalState	11
3.18	SIPPortsStatus	11
3.19	SS7Connections	12
3.20	Sla	12
3.21	SecurityStatus	13
3.22	SoftwareInventory	13
3.23	SoftwareVersionsInstalled	13



3.24	SoftwareVersionsRunning	14
3.25	SystemEnvironmentVariables	14
3.26	SystemStatus	14
3.27	UpgradeList	15
3.28	VirtualDicosProcessOutage	15
3.29	VmLogs	15
3.30	XdmsCaiLicence	15
3.31	XdmsInstance	16
3.32	XdmsRpm	16
3.33	XdmsTrafficApps	16
4	Health Check Profiles	17
4.1	HcMtasMandatory	17
4.2	HcMtasMediumPriority	17
4.3	HcMtasLowPriority	18
5	Problem Reporting	21



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 three profiles described in Section 4 on page 17.

The mandatory 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 mandatory profile is performed periodically once per hour. In troubleshooting situations or when more information is desired, the checks can be performed manually, optionally with a broader profile.

2.1 Packers List

To check which profiles are available use *cdclsv-list-packers* command. As a result the list of DNs is displayed:

```
cdclsPk=HcMtasLowPriority,cdcls=CDCLSVSite  
cdclsPk=HcMtasMandatory,cdcls=CDCLSVSite  
cdclsPk=HcMtasMediumPriority,cdcls=CDCLSVSite
```

Each DN denotes a reference to *cdclsv* pack object related to proper Health Check profile. Each *cdclsv* pack object has its own configuration and a list of checkers.

2.2 Manual Health Check

To perform selected profile use *cdclsv-pack <DN>*.

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

Execution status of selected DN can be checked by *cdclsv-pack-status <DN>*.

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

2.3 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.

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..

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

2.4 SLA Results

The SLA Results are stored under the Sla Directory.

The directory 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.

2.5

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.
!	FAIL	Problem detected by automatic health check.
E	ERROR	An error occurred, 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

OK	When there is one or more backup in the system, which can be restored.
VERIFY	When no backup is available for restoration.
FAIL	When an error has been found during data acquisition from backup manager.



3.4 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.

3.5 CpuLoadOnPLs

This step verifies if the CPU load on each PL is below the overload protection limit (85%).

Verdict

OK	If CPU load is less than the reference value by at least 10%.
VERIFY	If CPU load is closer to the reference value than 10%.
FAIL	If CPU load is higher than the reference value.

3.6 CpuLoadOnSCs

This step verifies if the CPU load on each SC is below the overload protection limit (85%).

Verdict

OK	If CPU load is less than the reference value by at least 10%.
VERIFY	If CPU load is closer to the reference value than 10%.
FAIL	If CPU load is higher than the reference value.

3.7 DiameterPortsStatus

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

Verdict



OK	Diameter stack is configured and at least one link is active.
VERIFY	(never)
FAIL	Diameter link is not configured or none of links is active.

3.8 DiskUsageOnSCs

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

Verdict

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.

Verdict

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

3.11 MemoryUsageOnPLs

This step checks whether memory usage is below the overload protection limit (95%) on each payload node.



Verdict

OK	If memory usage is less than the reference value by at least 10%.
VERIFY	If memory usage is closer to the reference value than 10%.
FAIL	If memory usage is higher than the reference value.

3.12 MemoryUsageOnSCs

This step checks whether memory use is below the overload protection limit (95%) on each system controller node.

Verdict

OK	If memory usage is less than the reference value by at least 10%.
VERIFY	If memory usage is closer to the reference value than 10%.
FAIL	If memory usage is higher than the reference value.

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



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 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.

3.18 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.

3.19 SS7Connections

This step verifies SS7 stack status.

Verdict

OK	If SS7 stack is configured correctly and the Application Servers are in proper state.
VERIFY	If SS7 stack is configured, but any of the Application Servers are in improper state.
FAIL	If no Application Server instances are found.
INFO	If SS7 stack is not configured.

3.20 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	The Verdict is OK when all the following conditions are fulfilled: <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	The Verdict is VERIFY when any of the following conditions are fulfilled: <ul style="list-style-type: none">• Any VM outage is detected.• If any VM has left the cluster and is not joined.
FAIL	The Verdict is FAIL when any of the following conditions are fulfilled: <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 2.4 SLA Results on page 4.

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

3.21 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.

3.22 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.

3.23 SoftwareVersionsInstalled

This step collects a list of software installed and running on the node.

Verdict

OK	(never)
VERIFY	(never)
FAIL	(never)
INFO	Returns a list of all the imported bundles and campaigns which are installed on the system.



3.24 SoftwareVersionsRunning

This step collects a list of software installed and running on the node.

Verdict

OK	(never)
VERIFY	(never)
FAIL	(never)
INFO	Returns a list of all the imported bundles and campaigns which are running on the system.

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



3.27 UpgradeList

This step based on Core MW repository list verifies if all campaigns are updated.

Verdict

OK	If all the campaigns are in committed status.
VERIFY	If any of the campaigns are not in committed status.
FAIL	(never)

3.28 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.

3.29 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.
FAIL	(never)

3.30 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.



3.31 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.

3.32 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.
VERIFY	(never)
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	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 HcMtasMandatory

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

HcMtasMandatory profile includes the following steps:

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

4.2 HcMtasMediumPriority

This profile provides a broader insight to the health of the system, by checking memory and CPU use, and some of the external interfaces.



Medium priority profile contains all steps from mandatory profile and the following steps:

- AllMtasPortsStatus
- CpuLoadOnPLs
- CpuLoadOnSCs
- DiameterPortsStatus
- DiskUsageOnSCs
- MemoryUsageOnPLs
- MemoryUsageOnSCs
- SIPPortsStatus
- XdmsCaiLicence
- XdmsRpm
- XdmsTrafficApps

Attention!

Running Health Check using this profile may cause CPU load peaks and increase of memory use.

4.3 HcMtasLowPriority

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

Low priority profile contains all steps included in the medium profile and the following steps:

- BackupList
- SecurityStatus
- SoftwareInventory
- SoftwareVersionsInstalled
- SoftwareVersionsRunning
- SystemEnvironmentVariables



- UpgradeList
- VmLogs

Attention!

Running Health Check using this profile with high system load may cause traffic disturbance.





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*.