

# Health Check Management

---

## DESCRIPTION

**Copyright**

© Ericsson AB 2015. 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

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Functions and Concepts</b>	<b>3</b>
2.1	Health Check Report File	4
2.2	Report File Housekeeping	7
2.3	Types of Operation	7
<b>3</b>	<b>Managed Object Model</b>	<b>13</b>
<b>4</b>	<b>Configuration Management</b>	<b>15</b>
<b>5</b>	<b>Security Management</b>	<b>17</b>





# 1 Introduction

This document provides an overview of the management model and concepts associated with the Health Check Management (HCM) managed area.

A managed area is represented by a group of Managed Object Classes (MOCs) within the Managed Object Model (MOM).





## 2 Functions and Concepts

HCM provides a management interface for reporting a summary of the Managed Element (ME) current health state highlighting any deviations from a normal behavior. It can be used to attend upgrades and to support preventive maintenance and problem resolution. The health status is obtained by verifying a set of rules.

A rule is a formal representation of a check. It contains information about what is checked: the command to be executed, the checks performed on the command printouts, and a recommendation about what to do if the result is not as expected. A rule, according to its definition, belongs to at least one category. A rule has an assigned severity (`CRITICAL` or `WARNING`) that helps understanding the severity if the check result of the rule fails. Optionally, a rule can be designed to accept input parameters through the model, that are used during the evaluation phase. A rule whose evaluation is performed using input parameters is a customizable rule, otherwise it is a simple rule. Input parameters of a rule allow the application node to customize, according to specific node characteristics, the rule check by specifying proper values to be used during the rule evaluation phase.

Rules are defined by the application node and written in XML format.

The evaluation of the health status is done through the execution of a health check job. It is only possible to execute one health check job at a time.

A health check job is created manually. It is associated to rule categories, executes all the installed rules belonging to the related categories, and computes the health status of the ME by checking the output of the rule commands against the defined evaluation criteria.

Possible ME statuses are as follows:

- `HEALTHY`: all executed rule checks are successful.
- `WARNING`: only rules whose severity is `WARNING` failed. In such a case, the ME requires attention.
- `NOT HEALTHY`: at least one rule whose severity is `CRITICAL` failed. In such a case, the ME requires immediate action to understand what caused the `CRITICAL` status and how to recover from it.

Health check job execution can be done in either of the following ways:

- **Manually**: the user launches job execution directly.
- **Automatically scheduled**: a time schedule is defined for job execution. The time schedule can be based on a single event (the date and time of job execution), on a periodic event (an event occurring at given intervals), and



on a calendar-based event (an event occurring at given intervals defined using events in the calendar).

- Triggered by another health check job: when the execution of a health check job detects the ME status to be different from `HEALTHY` it is possible to trigger the execution of a different job to perform further checks to collect more information about the ME state.

The ME status, as computed from the health check job, is available in the following ways:

- In the model, in proper Managed Object (MO) attribute.
- In a report file. It is located, by default, in a predefined directory under the `FileM` MO but it is possible to set a different user-defined one. The user-defined report file location can be either the URI of local directory or the SFTP URI of a remote destination. The job execution report file contains details about job execution such as ME status and information about rules execution, see Section 2.1 Health Check Report File on page 4 for details.

## 2.1 Health Check Report File

The result of a health check job execution is logged in a report file. It is an XML file available by default in a predefined directory on the `FileM` MO but it is possible to set a local or remote user-defined location. The report file contains information about the job, the result of executed rules, and computed ME status. It is composed of the following two sections:

- Header: Holds the job name, date when the job was executed, categories associated, and computed ME status.
- Body: Holds details about the executed rules.

The body has two subsections, one for failed rules and one for successful rules, where details about each executed rule are provided.

### 2.1.1 Report File XML Structure

A detailed description of the XML schema for the report file is shown in Figure 1.

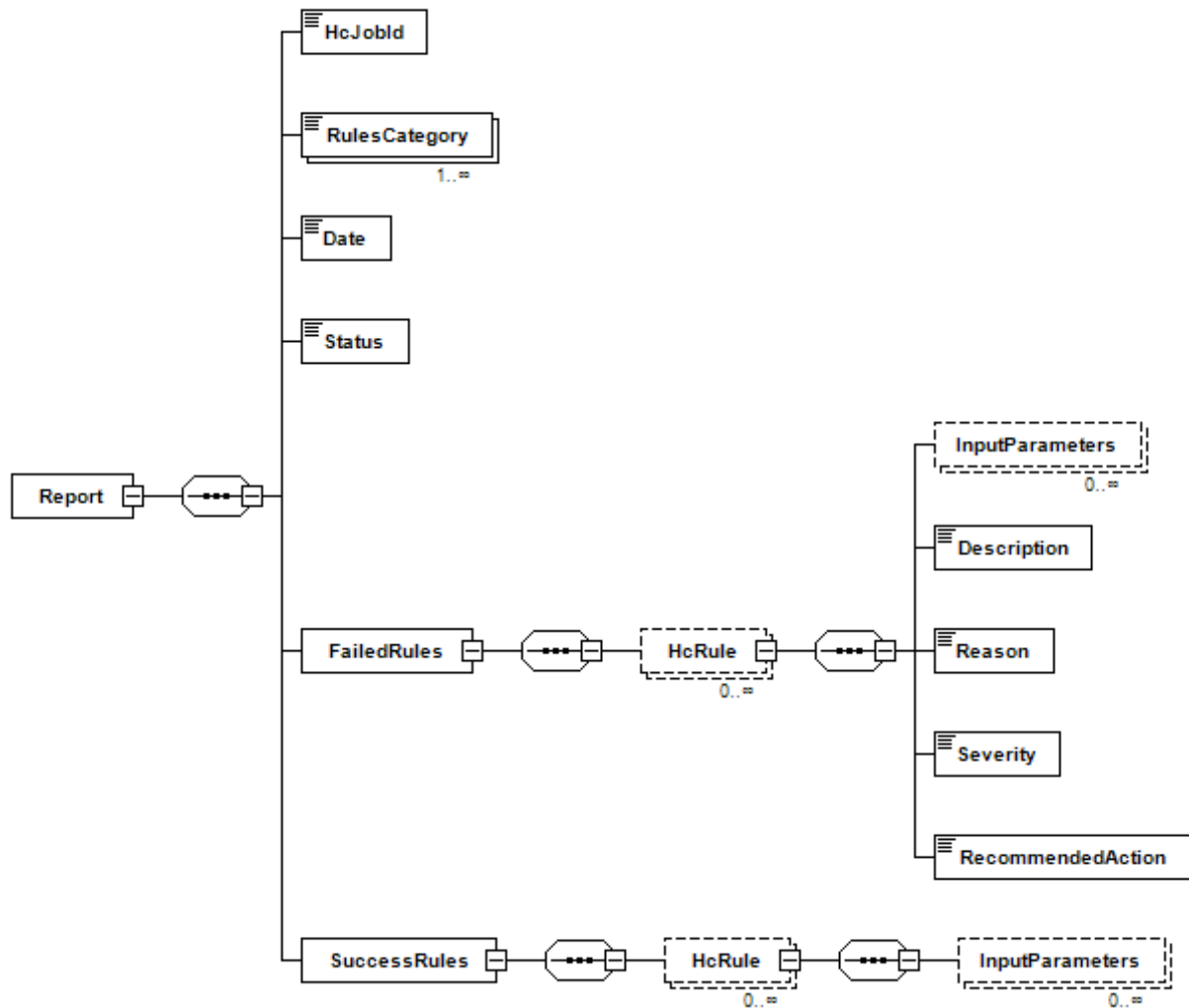


Figure 1 Schema Diagram for Report File

The report file starts with the tag <Report>. Only one such tag is present.

Under the <Report> tag, the children tags listed in Table 1 are present.

Table 1 <Report> Tag Children

Element Name	Cardinality	Description
HcJobId	1	This element provides the name of the job whose execution generated the report file.
RulesCategory	1..*	An instance of this tag is present for each rule category associated to the job whose execution generated the report file.



Element Name	Cardinality	Description
Date	1	The date and time of job execution.
Status	1	The ME state, as result of the health check job execution.
FailedRules	1	Contains information about the rules whose check evaluation failed during job execution.
SuccessRules	1	Contains information about the rules whose check evaluation succeeded during job execution.

The `<FailedRules>` and `<SuccessRules>` tags have the child tag listed in Table 2.

Table 2 `<FailedRules>` and `<SuccessRules>` Tag Children

Element Name	Cardinality	Description
HcRule	0..*	General information about the rule executed by the job.

The `<HcRule>` has two different definitions according to whether it is under the `<FailedRules>` tag or `SuccessRules` tag.

Under the `<FailedRules>` tag, the `<HcRule>` element has the attributes listed in Table 3.

Table 3 `<HcRule>` Attributes

Attribute Name	Mandatory	Description
id	yes	The identifier of the rule.
name	yes	The name of the rule.

The children tags are listed in Table 4.

Table 4 `<HcRule>` for `<FailedRules>` Children

Element Name	Cardinality	Description
Description	1	Contains a short description of the purpose of the rule.
Reason	1	Contains the error message to be shown when the rule result is not the expected one.



Element Name	Cardinality	Description
Severity	1	The severity of the rule.
RecommendedAction	1	Recommended action in case of check failure.
InputParameters	0..*	One of these tags is present for each, if any, user-defined value set for customizable rule.

Under the `<SuccessRules>` tag, the `<HcRule>` has the attributes listed in Table 3 and the children listed in Table 5.

Table 5 `<HcRule>` for `<SuccessRules>` Children

Element Name	Cardinality	Description
InputParameters	0..*	One of these tags is present for each, if any, user-defined value set for customizable rule.

## 2.2 Report File Housekeeping

The report file, reporting details about health check job execution, is produced upon successful job completion. A compressed archive file, containing logs used for the rules evaluation, is also provided, to be used for further investigation. Both the report file and the compressed archive are located in the same directory that can be either the default location under the `FileM MO` or a user-defined one.

For the report files stored in the default location under the `FileM MO`, HCM provides a housekeeping policy. That is, the maximum number of coupled report files with related compressed archive file is per default set to 10. A different value can be set by the user. If a new coupled report file and archive file are produced, exceeding the maximum number, the oldest couple is deleted.

## 2.3 Types of Operation

HCM supports the following operations:

- List health check rules

This operation lists all the rules installed on the ME. Details such as name of the rule, related categories, description, recommended action in case of failure, and severity are provided. For further details on how to perform this operation, refer to *List Health Check Rules*.



- Create a health check job

This operation creates a health check job with the specified name. At least one rule category is associated to the job. It is possible to set the name of another job to be triggered in case the ME status is different from `HEALTHY`. For further details on how to perform this operation, refer to *Create Health Check Job*.

- Lock health check rule

This operation instructs how to lock a rule. If a rule is locked, it is skipped by job execution. By default, all rules are unlocked. For further details on how to perform this operation, refer to *Lock Health Check Rule*.

- Unlock health check rule

This operation instructs how to unlock a rule. If a rule is unlocked, it is executed by jobs defined to execute rules belonging to the rule category. By default, all rules are unlocked. For further details on how to perform this operation, refer to *Unlock Health Check Rule*.

- Modify health check rule parameter

This operation modifies values for parameters used when a health check rule is evaluated. Default values are used if no changes are performed. The modification can be done before executing a specific health check job, so it applies only for that job. The possibility to modify the criteria used for a rule evaluation is defined at node installation time. For further details on how to perform this operation, refer to *Modify Health Check Rule Parameter*.

- Restore default value for health check rule parameter

This operation restores default values for parameters used when a health check rule is evaluated. The operation can be done before executing a specific health check job, so it applies only for that job. The possibility to modify the criteria used for a rule evaluation is defined at node installation time. For further details on how to perform this operation, refer to *Restore Default Value for Health Check Rule Parameter*.

- Execute a health check job

This operation executes a previously created health check job. All the rules installed on the machine, and belonging to the rule categories associated to the job, are executed. The related checks are performed against the evaluation criteria, and, as a result, the health status of the ME is computed. The ME status is `HEALTHY` if all rule checks are successful. If at least one rule whose severity is critical fails, the status is set to `NOT HEALTHY`. If only rules whose severity is warning fail, the status is set to `WARNING`.

The result of health check job execution is available through the attribute `status` on the `HcJob` MO, and in a report file. For details about report file structure, see Section 2.1 Health Check Report File on page 4. The result file is stored in a predefined directory under the `FileM` MO, together with



a compressed archive file containing logs used for the rule evaluation. It is possible to specify a user-defined location that can be either the URI of local directory or the SFTP URI of a remote destination. The user-defined location for report file and compressed archive is also applied to the triggered job, if any.

**Note:** In the predefined directory under the `FileM` MO, the maximum number of couple report file and related archive file containing logs allowed is per default 10. This threshold can be configured. If a new couple is to be produced, exceeding the threshold, the oldest couple is deleted.

Only the active installed rules are executed; the locked ones are skipped by the job execution. For details on how to lock or unlock a rule, refer to *Lock Health Check Rule* and *Unlock Health Check Rule*.

Details about the health check job execution progress are provided to monitor it until completion.

For further details on job execution, storing report file in default location, and how to check the job execution progress, refer to *Execute Health Check Job*.

For further details on job execution, storing report file in user-defined location, and how to check the job execution progress, refer to *Execute Health Check Job Providing Export URI*.

- List available health check jobs

This operation lists available health check jobs created on the ME. Details such as associated categories, last computed ME status, local default path of the report file, name of the last report file, and rules failed during last execution are provided for each present job. For further details on how to perform this operation, refer to *List Health Check Jobs*.

- Modify health check job to trigger

This operation allows the user to modify, for a specific health check job, the name of the job to be triggered if the first job execution detects an ME status different from `HEALTHY`. The following can be done:

- Set job to trigger: for jobs having no previously set job to trigger
- Modify job to trigger: for jobs having a previously set job to trigger

For further details on how to perform these operations, refer to *Modify Health Check Job To Trigger*.

- Remove health check job to trigger

This operation allows the user to delete, for a specific health check job, the name of previously set job to be triggered. Once this operation



is completed, if the job execution detects an ME status different from `HEALTHY` no other job execution is performed.

For further details on how to perform this operation, refer to *Remove Health Check Job To Trigger*.

- Delete health check job

This operation deletes a health check job. For further details on how to perform this operation, refer to *Delete Health Check Job*.

- Schedule health check job

This operation instructs how to schedule a health check job to start automatically at a specified time. The following three scheduling policies are available:

- Calendar-based periodic event: a health check job is executed regularly at a desired interval using calendar events, during a time slice. By default, the time slice starts at the current system time when defining the periodic event and ends at the end of the century, but can be configured by the user. The event is defined specifying the time and, optionally, the day of month, day of week, day of week occurrence, and month (for example, at 13:30 pm on the second day of March falling on Monday).
- Periodic event: a health check job is executed regularly at a specified time interval during a time slice. By default, the time slice starts at the current system time when defining the periodic event and ends at the end of the century but can be configured by the user. The time interval is defined by the number of hours and optionally months, weeks, days, and minutes between two consecutive health check job executions (for example, once every 12 hours).
- Single event: a health check job is executed once at a specified date and time (for example, on the 15th of April 2015 at 15:15:00).

The scheduler can be unlocked to enable periodic job execution or locked to disable it. By default the scheduler is locked, so it is necessary to unlock the scheduler to make scheduling policy effective. A scheduled event can be deleted when no longer needed.

For further details on how to schedule jobs according to each available policy, refer to the following documents:

- *Schedule Single Health Check Job*
- *Schedule Health Check Job Based on Calendar Event*
- *Schedule Health Check Job Based on Periodic Event*



For a description of how to lock or unlock a scheduler event and how to delete a scheduled execution, refer to the following documents:

- *Lock Health Check Job Scheduler*
- *Unlock Health Check Job Scheduler*
- *Delete Scheduled Event*



### 3 Managed Object Model

The HCM managed area is represented in the *Managed Object Model (MOM)* as follows:

```
ManagedElement
+-SystemFunctions
+-HealthCheckM
+-HcJob
+-HcJobScheduler
+-HcCalendarBasedPeriodicEvent
+-HcPeriodicEvent
+-HcSingleEvent
+-HcRule
```

For general information about the MOM, MOCs, MOs, cardinality, and related concepts, refer to *Managed Object Model User Guide*.

The HCM MOCs are described in Table 6.

**Table 6 HCM Managed Object Class Descriptions**

Managed Object Class	Description
<i>HealthCheckM</i>	<p>The root of the HCM model.</p> <p>The <code>maxNoOfReportFiles</code> attribute holds the maximum amount of coupled report file and related compressed archive logs allowed in the default output directory under the <code>FileM</code> MO. It is used to manage the report file housekeeping policy for the default output file location under the <code>FileM</code> MO.</p>
<i>HcJob</i>	<p>Represents a health check job that can be executed on the ME.</p> <p>The <code>progressReport</code> attribute is intended to report details about the job execution progress.</p>



Table 6 HCM Managed Object Class Descriptions

Managed Object Class	Description
<i>HcRule</i>	<p>Represents the health check rule. It contains detailed information necessary to define each specific health check rule.</p> <p>The <i>HcRule</i> MO specifies the following:</p> <ul style="list-style-type: none"><li>• The rule severity: each rule has an associated severity that can be warning or critical</li><li>• The rule category: each rule belongs to at least one health check category</li><li>• Default values for parameters of customizable rules.</li></ul>
<i>HcJobScheduler</i>	<p>Represents the health check job scheduler to, at a specified time, start a job automatically. The following three scheduling policies are supported:</p> <ul style="list-style-type: none"><li>• Calendar-based periodic event</li><li>• Periodic event</li><li>• Single event</li></ul>
<i>HcCalendarBasedPeriodicEvent</i>	Represents a calendar-based event for executing a health check job periodically.
<i>HcPeriodicEvent</i>	Represents a time interval-based event for executing a health check job periodically.
<i>HcSingleEvent</i>	Represents a date or time-based event for executing a health check job once.



## 4 Configuration Management

HCM is accessed using NETCONF or the Ericsson Command-Line Interface (ECLI) to manipulate the Management Information Base (MIB).

The following operations can be performed by the user and are described in Operating Instructions using the ECLI:

### **Basic Health Check Operations**

- *List Health Check Rules*
- *Create Health Check Job*
- *Lock Health Check Rule*
- *Unlock Health Check Rule*
- *Modify Health Check Rule Parameter*
- *Restore Default Value for Health Check Rule Parameter*
- *Execute Health Check Job*
- *Execute Health Check Job Providing Export URI*
- *List Health Check Jobs*
- *Modify Health Check Job To Trigger*
- *Remove Health Check Job To Trigger*
- *Delete Health Check Job*

### **Schedule Health Check Job**

- *Schedule Single Health Check Job*
- *Schedule Health Check Job Based on Calendar Event*
- *Schedule Health Check Job Based on Periodic Event*

### **Manage Scheduled Health Check Jobs**

- *Lock Health Check Job Scheduler*
- *Unlock Health Check Job Scheduler*
- *Delete Scheduled Event*





## 5 Security Management

HCM access is managed by an authentication and authorization mechanism. For each HCM role, specific rules are applied to determine the scope of what is accessible.

The following two HCM roles are defined:

- `SystemAdministrator`
- `SystemReadOnly`

Once authenticated as a `SystemAdministrator`, full access (read, write, execute) is granted to the *HealthCheckM* MO and its attributes and actions.

Once authenticated as a `SystemReadOnly`, read access is granted to the *HealthCheckM* MO and its attributes.

**Note:** If the `SystemReadOnly` role is not defined, the related access rule is not installed.