|  |  |
| --- | --- |
| 3GPP TS 28.111 18.0.0 (2024-03) | |
| Technical Specification | |
| 3rd Generation Partnership Project;  Technical Specification Group Services and System Aspects;  Management and orchestration;  Fault Management (FM)  (Release 18) | |
|  | |
|  |  |
|  | |
| The present document has been developed within the 3rd Generation Partnership Project (3GPP TM) and may be further elaborated for the purposes of 3GPP. The present document has not been subject to any approval process by the 3GPPOrganizational Partners and shall not be implemented. This Specification is provided for future development work within 3GPPonly. The Organizational Partners accept no liability for any use of this Specification. Specifications and Reports for implementation of the 3GPP TM system should be obtained via the 3GPP Organizational Partners' Publications Offices. | |

|  |
| --- |
|  |
| ***3GPP***  Postal address  3GPP support office address  650 Route des Lucioles - Sophia Antipolis  Valbonne - FRANCE  Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16  Internet  http://www.3gpp.org |
| ***Copyright Notification***  No part may be reproduced except as authorized by written permission. The copyright and the foregoing restriction extend to reproduction in all media.  © 2024, 3GPP Organizational Partners (ARIB, ATIS, CCSA, ETSI, TSDSI, TTA, TTC).  All rights reserved.  UMTS™ is a Trade Mark of ETSI registered for the benefit of its members  3GPP™ is a Trade Mark of ETSI registered for the benefit of its Members and of the 3GPP Organizational Partners LTE™ is a Trade Mark of ETSI registered for the benefit of its Members and of the 3GPP Organizational Partners  GSM® and the GSM logo are registered and owned by the GSM Association |

Contents

Foreword 5

1 Scope 7

2 References 7

3 Definitions of terms, symbols and abbreviations 8

3.1 Terms 8

3.2 Symbols 8

3.3 Abbreviations 8

4 Concepts and overview 8

5 Requirements 9

6 Solution description 10

6.1 Solution components 10

6.2 Model driven approach 11

6.3 Alarm records 11

6.4 Alarm identification 12

6.5 Alarm lists 12

6.6 Retrieving alarm records by MnS consumers 12

6.7 Acknowledging alarms by MnS consumers 12

6.8 Clearing alarms by MnS consumers 13

6.9 Commenting alarms by MnS consumers 13

6.10 Alarm correlation 13

6.11 Reliability of alarm lists 13

6.12 Alarm notifications 13

6.13 Alarm list states 14

6.14 Alarm record life cycle 15

7 Model 15

7.1 Imported information entities and local labels 15

7.2 Class diagrams 16

7.2.1 Relationships 16

7.2.2 Inheritance 16

7.3 Class definitions 16

7.3.1 AlarmRecord <<dataType>> 16

7.3.1.1 Definition 16

7.3.1.2 Attributes 17

7.3.1.3 Attribute constraints 17

7.3.1.4 Notifications 17

7.3.2 AlarmList 18

7.3.2.1 Definition 18

7.3.2.2 Attributes 18

7.3.2.3 Attribute constraints 18

7.3.2.4 Notifications 18

7.3.3 AlarmComment <<dataType>> 18

7.3.3.1 Definition 18

7.3.3.2 Attributes 18

7.3.3.3 Attribute constraints 19

7.3.3.4 Notifications 19

7.3.4 CorrelatedNotification <<dataType>> 19

7.3.4.1 Definition 19

7.3.4.2 Attributes 19

7.3.4.3 Attribute constraints 19

7.3.4.4 Notifications 19

7.4 Attribute definitions 19

7.4.1 Attribute properties 19

7.4.2 Constraints 25

7.5 Common notifications 26

7.5.1 Alarm notifications 26

7.5.2 Configuration notifications 26

8 Notifications 26

8.1 Overview 26

8.2 notifyNewAlarm 26

8.2.1 Definition 26

8.2.2 Input parameters 26

8.3 notifyClearedAlarm 27

8.3.1 Definition 27

8.3.2 Input parameters 28

8.4 notifyChangedAlarmGeneral 28

8.4.1 Definition 28

8.4.2 Input parameters 29

8.5 notifyAlarmListRebuilt 29

8.5.1 Definition 29

8.5.2 Input parameters 30

8.6 notifyChangedAlarm 30

8.6.1 Definition 30

8.6.2 Input parameters 31

8.7 notifyCorrelatedNotificationChanged 31

8.7.1 Definition 31

8.7.2 Input parameters 31

8.8 notifyAckStateChanged 31

8.8.1 Definition 31

8.8.2 Input parameters 32

8.9 notifyComments 32

8.9.1 Definition 32

8.9.2 Input parameters 32

8.10 notifyPotentialFaultyAlarmList 32

8.10.1 Definition 32

8.10.2 Input parameters 33

Annex A (normative): Solution sets 34

A.1 RESTful HTTP-based solution set 34

A.1.1 Mapping of the NRM 34

A.1.2 Mapping of notifications 34

A.1.3 OpenAPI definitions 34

A.1.4 Examples 34

A.2 RESTful HTTP-based solution set for integration with ONAP VES API 36

A.2.1 General 36

A.2.2 Mapping of notifications 36

A.2.2.1 General 36

A.2.2.2 Resources 36

A.2.3 Integration with ONAP VES 37

A.3 NETCONF/YANG solution set 37

A.3.1 General 37

A.3.2 YANG definitions 37

Annex B (informative): Probable Causes 38

Annex C (informative): Change history 44

# Foreword

This Technical Specification has been produced by the 3rd Generation Partnership Project (3GPP).

The contents of the present document are subject to continuing work within the TSG and may change following formal TSG approval. Should the TSG modify the contents of the present document, it will be re-released by the TSG with an identifying change of release date and an increase in version number as follows:

Version x.y.z

where:

x the first digit:

1 presented to TSG for information;

2 presented to TSG for approval;

3 or greater indicates TSG approved document under change control.

y the second digit is incremented for all changes of substance, i.e. technical enhancements, corrections, updates, etc.

z the third digit is incremented when editorial only changes have been incorporated in the document.

In the present document, modal verbs have the following meanings:

**shall** indicates a mandatory requirement to do something

**shall not** indicates an interdiction (prohibition) to do something

The constructions "shall" and "shall not" are confined to the context of normative provisions, and do not appear in Technical Reports.

The constructions "must" and "must not" are not used as substitutes for "shall" and "shall not". Their use is avoided insofar as possible, and they are not used in a normative context except in a direct citation from an external, referenced, non-3GPP document, or so as to maintain continuity of style when extending or modifying the provisions of such a referenced document.

**should** indicates a recommendation to do something

**should not** indicates a recommendation not to do something

**may** indicates permission to do something

**need not** indicates permission not to do something

The construction "may not" is ambiguous and is not used in normative elements. The unambiguous constructions "might not" or "shall not" are used instead, depending upon the meaning intended.

**can** indicates that something is possible

**cannot** indicates that something is impossible

The constructions "can" and "cannot" are not substitutes for "may" and "need not".

**will** indicates that something is certain or expected to happen as a result of action taken by an agency the behaviour of which is outside the scope of the present document

**will not** indicates that something is certain or expected not to happen as a result of action taken by an agency the behaviour of which is outside the scope of the present document

**might** indicates a likelihood that something will happen as a result of action taken by some agency the behaviour of which is outside the scope of the present document

**might not** indicates a likelihood that something will not happen as a result of action taken by some agency the behaviour of which is outside the scope of the present document

In addition:

**is** (or any other verb in the indicative mood) indicates a statement of fact

**is not** (or any other negative verb in the indicative mood) indicates a statement of fact

The constructions "is" and "is not" do not indicate requirements.

# 1 Scope

The present document describes the SBMA based Fault Management service (see [14]). It includes stages 1, 2 and 3.

The present document of the Fault Management MnS is based on the SBMA principles using CRUD operations, modeled OAM data in the NRM together with fault management specific notifications. An IRP based solution for fault management is out of scope for the present document.

# 2 References

The following documents contain provisions which, through reference in this text, constitute provisions of the present document.

- References are either specific (identified by date of publication, edition number, version number, etc.) or non‑specific.

- For a specific reference, subsequent revisions do not apply.

- For a non-specific reference, the latest version applies. In the case of a reference to a 3GPP document (including a GSM document), a non-specific reference implicitly refers to the latest version of that document *in the same Release as the present document*.

[1] 3GPP TR 21.905: "Vocabulary for 3GPP Specifications".

[2] 3GPP TS 28.532: "Management and orchestration; Generic Management services".

[3] ETSI TS 101 251 (V6.3.0): "Digital cellular telecommunications system (Phase 2+); Fault management of the Base Station System (BSS) (GSM 12.11 version 6.3.0 Release 1997)".

[4] 3GPP TS 28.516: "Fault Management (FM) for mobile networks that include virtualized network functions; Procedure".

[5] 3GPP TS 28.622: "Telecommunication management; Generic Network Resource Model (NRM) Integration Reference Point (IRP); Information Service (IS)".

[6] ITU-T Recommendation X.721 (02/92): "Information technology - Open Systems Interconnection - Structure of management information: Definition of management information".

[7] ITU-T Recommendation M.3100: "Generic network information model".

[8] ITU-T Recommendation X.733 (02/92): "Information technology - Open Systems Interconnection - Systems Management: Alarm reporting function".

[9] Text Attribution: Creator: ONAP, under Creative Commons Attribution 4.0 International License, https://creativecommons.org/licenses/by/4.0/, URI to access the text: <https://github.com/onap/vnfrqts-requirements/blob/05f26fac2b941513a7d0e856b99fd8c61d688299/docs/Chapter8/ves7_1spec.rst#resource-structure>.

[10] 3GPP TS 32.158: "Management and orchestration; Design rules for Representational State Transfer (REST) Solution Sets (SS)".

[11] Management and Orchestration APIs Stage 3 Repository <https://forge.3gpp.org/rep/sa5/MnS/-/tree/Tag_Rel18_SA103/>.

[12] 3GPP TS 32.401: "Telecommunication management; Performance Measurement (PM); Concept and requirements".

[13] ITU-T Recommendation X.736 (01/92): "Information technology - Open Systems Interconnection - Systems Management: Security alarm reporting function".

[14] 3GPP TS 28.533: "Management and orchestration; Architecture framework".

[15] 3GPP TS 32.160: "Management and orchestration; Management service template".

# 3 Definitions of terms, symbols and abbreviations

## 3.1 Terms

For the purposes of the present document, the terms given in 3GPP TR 21.905 [1] and the following apply. A term defined in the present document takes precedence over the definition of the same term, if any, in 3GPP TR 21.905 [1].

**Alarm:** A representation of an error or failure that requires attention or reaction by an operator or some machine. Alarms have state.

**Alarm identifying attributes:** A set of attributes (*objectInstance, alarmType, probableCause and specificProblem*, if present) that identify an alarm. *ObjectInstance* identifies the network resource ,while *alarmType, probableCause* and *specificProblem* (if present) identify the alarming condition.

**Error:** A state of the system different from the correct system state. An error may or may not lead to a service failure. An error has a begin and end time.

**Event:** Anything that occurs at a certain point in time, for example a configuration change, a threshold crossing, a transition to an error state or a transition to a failure state. Events do not have states.

**Failure:** A state of inability to deliver the correct service as defined by the service specification. A service failure is the result of an error. A failure has a begin and end time.

**Fault:** The (hypothesized or adjudged) cause for an error or a failure (such as system malfunctions, a defect in system design, a defect in software, or external interference).

**MonitoredEntity:** Any class that can have an alarmed state.

**Root cause:** The primary fault (cause), if any, leading to one or multiple errors or failures.

## 3.2 Symbols

Void.

## 3.3 Abbreviations

For the purposes of the present document, the abbreviations given in 3GPP TR 21.905 [1] and the following apply. An abbreviation defined in the present document takes precedence over the definition of the same abbreviation, if any, in 3GPP TR 21.905 [1].

ADAC Automatically Detected and Automatically Cleared

ADMC Automatically Detected and Manually Cleared

CRUD Create, Read, Update, Delete basic data manipulation operations

FM Fault Management

ME Managed Element

MnS Management Service

NRM Network Resource Model

# 4 Concepts and overview

A (managed) systems may experience faults such as malfunctions, a defect in system design, a defect in the software, or external interference. These faults may (or may not) lead to a system state that is different from the correct or desired system state. An incorrect system state is called error. Errors are hence caused by faults. Faults and errors are not always externally observable and may remain undetected.

Errors, in turn, may (or may not) cause failures. A failure is the inability to deliver the correct service as defined by the service specification. A failure is hence always externally observable.

In summary, a fault may case one or more errors, and an error may cause one or more failures.

An alarm is the management representation of a fault, a (detected) error or a failure that requires attention or reaction by an operator or some machine.

Fault Management is concerned with representing, managing, and reporting alarms. Fault Management is often also referred to as Alarm Management. The alarm model is independent from the underlying managed system. The same model can be used to represent alarms from any 3GPP generation or other networks and any resource. Specifics of the managed system manifest themselves only in the values of the information elements of the alarm model.

Alarms allow to report any kind of issue, from small faults without service impact to large scale failures of telecommunication services affecting many users.

A prerequisite for Fault Management as defined in the present document is that the managed system is represented by managed objects, that are organized in hierarchical object trees, in the management system.

The solution specified in the present document is based on ITU-T X.733 [8].

Fault Management is considered a generic management service. It shall be able to support fault indications about any generation of 3GPP or other networks and any resource that can be addressed by a distinguished name e.g. ManagedElements, ENBs or NetworkSlices or non-3GPP managed resources.

Fault management can handle alarms about any kind of fault in a 3GPP system from small hardware errors to service failures effecting many users.

# 5 Requirements

| Requirement label | Description | Motivation |
| --- | --- | --- |
| **REQ-FM-MC-1** | The 3GPP management system shall have the capability to provide **alarm notifications** to authorized consumers. | Motivation: the consumer should receive information about alarms immediately when an alarm is raised or changed. |
| **REQ-FM-MC-2** | The 3GPP management system shall have the capability to allow authorized consumers to **subscribe** to alarm notifications. | Motivation: Needed for REQ-FM-MC-1. Producers will not send notification without an explicit subscription. |
| **REQ-FM-MC-3** | The 3GPP management system shall have the capability to allow authorized consumers to **unsubscribe** from alarm notifications. | Motivation: The consumer needs to be able to indicate that it is no longer interested in receiving immediate alarm information |
| **REQ-FM-MC-4** | The 3GPP management system should have the capability to allow authorized consumers to provide a **filter** for alarm **notifications**. | Motivation: The consumer shall be able to indicate that it is interested only in a subset of alarms. |
| **REQ-FM-MC-5** | The 3GPP management system shall have the capability to allow authorized consumers to **retrieve the alarm list**. | Motivation: The consumer shall be able to read all current alarms. It needs this if the sequence of received alarm notifications does not provide a reliable and complete view of the alarm situation. This may happen after the start-up of the consumer fault management service, if the connection or some alarm notifications are lost, or if the alarm producer was not able to provide on-time indication of all alarm changes. |
| **REQ-FM-MC-6** | The 3GPP management system should have the capability to allow authorized consumers to **retrieve a filtered** subset of the **alarm list**. | Motivation: If the consumer is interested only in a subset of alarms, it shall be able to retrieve only that subset. |
| **REQ-FM-MC-7** | The 3GPP management system shall have the capability to provide **changed alarm notifications** to its authorized consumer. | Motivation: the consumer should receive information about changed alarms immediately. |
| **REQ-FM-MC-8** | The 3GPP management system shall have the capability to provide **cleared alarm notifications** to its authorized consumer. | Motivation: the consumer should receive information about cleared alarms immediately. |
| **REQ-FM-MC-9** | The 3GPP management system shall have the capability to provide **new** generated **alarm notifications** to its authorized consumer. | Motivation: the consumer should receive information about alarms immediately when an alarm is raised. |
| **REQ-FM-MC-10** | The 3GPP management system should have the capability to indicate that the **alarm list** is **potentially faulty**. | Motivation: the consumer should receive information when the alarm list is corrupt or out-of-date. |
| **REQ-FM-MC-11** | The 3GPP management system shall have the capability to indicate that the **alarm list** was **rebuilt** and is reliable again after a previous disturbance. | Motivation: the consumer should receive information when the correct alarm information is available again. |
| **REQ-FM-MC-12** | The 3GPP management system should have the capability to satisfy the request to **acknowledge** one or multiple **alarms**. If this capability is not supported, then the producer shall be able to automatically acknowledge alarms. | Motivation: the consumer should be able to register in the producer that it has received the alarm and has done some vendor specific level of processing of the alarm information. |
| **REQ-FM-MC-13** | The 3GPP management system should have the capability to satisfy the request to **clear** one or multiple **alarms**. This capability is only applicable if one or more of the alarms supported by the producer is of type ADMC. | Motivation: If the producer supports ADMC alarms, the consumer shall be able to clear those. |
| **REQ-FM-MC-14** | The 3GPP management system should have the capability to provide **acknowledgement** state change **notifications** to its authorized consumer. | Motivation: the consumer should receive information about acknowledged alarms immediately. |

# 6 Solution description

## 6.1 Solution components

The solution consists of the basic solution and the following optional solution components:

- Dedicated perceived severity change notification

- Acknowledging alarms by MnS consumers

- Commenting alarms by MnS consumers

- Alarm correlation

- Reliability of alarm lists

Table 6.1-1: FM solution components

| **Usage** | **Operations and notifications** | **NRM** |
| --- | --- | --- |
| FM basic | notifyNewAlarm  notifyChangedAlarmGeneral  notifyClearedAlarm | AlarmList |
| Dedicated perceived severity change notification | notifyChangedAlarm |  |
| Acknowledging alarms by MnS consumers | notifyAckStateChanged | alarmRecord.ackTime alarmRecord.ackUserId  alarmRecord.ackSystemId  alarmRecord.ackState |
| Commenting alarms by MnS consumers | notifyComments | alarmRecord:comments, datatype:alarmComment |
| Alarm correlation | notifyCorrelatedNotificationChanged | alarmRecord:correlatedNotifications  alarmRecord:rootCauseIndicator |
| Reliability of alarm lists | notifyPotentialFaultyAlarmList  notifyAlarmListRebuilt | AlarmList.unreliableAlarmScope |

## 6.2 Model driven approach

The solution for Fault Management is based on the model driven approach.

NRM data is written to control the behaviour of the fault management.

Data provided to the fault management consumer is made available in two ways (representing the same information). MnS consumers may use the a read operation to read any data. Additionally, data that should be provided as soon as it is available in the MnS producer is sent to subscribed MnS consumers in notifications (e.g. information about a new alarm).

For this reason, only an alarm model is defined. The CRUD operations defined in TS 28.532 [2], clause 11.1 are used for interacting with the instantiation of the model.

Since the generic provisioning notifications defined in TS 28.532 [2], clause 11.1 are not used in all cases, the present document also defines some specific alarm notifications to report changes in the alarm model.

Interactions with the alarm model with both operations and notifications may be subject to access control.

## 6.3 Alarm records

An alarm is described by a set of attributes. This set of attributes is referred to as alarm record. An alarm record is hence the management representation of an alarm.

The object instance attribute in an alarm record identifies the object that represents the alarmed entity in the management system. Objects are identified using their Distinguished Name (DN). Note that all is needed is a DN. It is not required that the object really exists in the management system and can be accessed with CRUD operations.

The alarm type (ITU-T X.733 [8], clause 8.1.1) attribute specifies roughly in which area of the supervised system an alarm has occurred:

- If the alarm type is equal to "COMMUNICATIONS\_ALARM", the alarm is principally associated with the procedures and/or processes required to convey information from one point to another.

- If the alarm type is equal to "PROCESSING\_ERROR\_ALARM", the alarm is principally associated with a software or processing fault.

- If the alarm type is equal to "EQUIPMENT\_ALARM", the alarm is principally associated with an equipment fault.

- If the alarm type is equal to "ENVIRONMENTAL\_ALARM", the alarm type is principally associated with a condition relating to an enclosure in which the equipment resides.

The present document also provides the alarm type "QUALITY\_OF\_SERVICE\_ALARM". This alarm type does not specify the area where the issue occurs but conveys that the alarm is principally associated with a degradation in the quality of a service. Also, this alarm type can be combined with any perceived severity. An alarm with this type is often generated, in addition to an alarm with one of the other types, for the same underlying fault. This allows to filter on alarms that are related to a (potential) service degradation only.

The specific problem attribute (ITU-T X.733 [8], clause 8.1.2.2) provides further refinements to the probable cause of the alarm.

The perceived severity attribute (ITU-T X.733 [8], clause 8.1.2.3) allows to assess the severity of the alarm condition as determined by the system. The values critical, major, minor and warning are provided, and the value cleared indicates that the condition leading to an alarm is not present anymore.

## 6.4 Alarm identification

Alarms with the same values for the attributes object instance, alarm type, probable cause and specific problem are considered the same alarm. These four attributes are also called alarm identifying attributes. As a shortcut for the alarm identifying attributes the alarm identifier is defined. To refer to a specific alarm it is hence possible to use the four alarm identifying attributes or the alarm identifier.

## 6.5 Alarm lists

The alarm records representing the current state of the system are stored in alarm lists on MnS producers. An alarm list contains the alarm records related to a certain management scope. This scope is either a manged element or a subnetwork. Historical alarm records are not stored in an alarm list. Therefore, at any point in time, there cannot be more than one alarm record in an alarm list, where the alarm identifying attributes have the same values.

Alarm lists are typically created automatically upon system start up. They cannot be created or deleted by MnS consumers.

The alarm records in the alarm list are created and deleted by the system. A MnS consumer can only read the attributes of alarm records but not manipulate them (except for a few exceptions).

Besides the alarm records itself, alarm lists contain also attributes describing the alarm records, such as the total number of alarm records in the alarm list or the time when an alarm record was updated the last time.

## 6.6 Retrieving alarm records by MnS consumers

A MnS consumer can retrieve the alarm records in an alarm list using the "getMOIAttributes" operation defined in TS 28.532 [2], clause 11.1.1.2. Often it is desired to retrieve only alarm records matching some criteria and not all alarms in an alarm list. For example, a MnS consumer might be interested only in alarms whose perceived severity is critical or in alarms from a specific managed element. This requires support for conditional data node retrieval.

## 6.7 Acknowledging alarms by MnS consumers

An alarm is defined as a fault, an error or failure that requires attention or reaction by an operator or some machine. For that reason, alarm records should not be removed from the alarm list without prior acknowledgement by the operator or a machine. The acknowledgement state attribute is provided for that purpose in an alarm record. It can have the values acknowledged and unacknowledged and is set by the MnS consumer.

When a new alarm record is created by the system, its acknowledgement state is set to unacknowledged. To acknowledge an alarm, a MnS consumer can set the attribute to acknowledged. A MnS consumer may also set back the state of a previously acknowledged alarm to unacknowledged. The MnS consumer may provide its identity (user identifier and system identifier) to the MnS Producer when setting the acknowledgement state attribute. The MnS Producer stores this information in the corresponding alarm record.

The system automatically captures the time when the acknowledgement state attribute is updated. A dedicated acknowledgement time attribute is provided for that purpose.

For reporting changes of the acknowledgement state refer to clause 6.12.

The possibility to acknowledge alarms is an optional feature.

## 6.8 Clearing alarms by MnS consumers

If the condition leading to an alarm is not prevailing or not detected anymore, the perceived severity of the alarm is set to cleared by the system. These alarms are referred to as automatically detected automatically cleared alarms (ADAC alarms). There are also alarms that are not automatically cleared. These alarms are referred to as automatically detected manually cleared alarms (ADMC alarms).

MnS consumers need to manually clear ADMC alarms by setting the perceived severity attribute of the alarm record to cleared. The MnS consumer may provide its identity (user identifier and system identifier) to the MnS producer when setting the attribute. The MnS Producer stores this information in the corresponding alarm record. If the fault condition still prevails, the system will create a new alarm or change the perceived severity value back to the old value, depending on if the alarm was removed or not removed after clearing it.

It is out of scope of the present document how the MnS consumer can find out if an alarm is an ADAC or ADMC alarm. Furthermore, it is outside the scope of the present document how a MnS consumer can find out that the fault condition does not exist anymore.

The possibility to clear alarms is a mandatory feature in case ADMC alarms may be raised by the system.

## 6.9 Commenting alarms by MnS consumers

A MnS consumer can add one or more comments, in the format of free text, to an alarm record. The MnS consumer may provide its identity (user identifier and system identifier) when adding a comment. Each comment is annotated automatically with the time it is created.

A MnS consumer cannot update or delete a comment. Comments are deleted automatically when the corresponding alarm record is deleted.

For reporting the addition of a comment refer to clause 6.12.

The possibility to comment alarms is an optional feature.

## 6.10 Alarm correlation

Multiple errors and failures may be caused by a single fault. A single error may result also in multiple failures. The system may support identifying these relationships between faults, errors, and alarms.

To capture these relationships the correlated notifications attribute and the root cause indicator attribute are provided. Modifications of these attributes are reported using the notify correlated notification changed notification.

## 6.11 Reliability of alarm lists

Alarm lists may become unreliable for numerous reasons. Due to the organisation of managed objects (that can be alarmed and have related alarm records in the alarm list) in hierarchical object trees, alarm records relating to a complete subtree are typically becoming unreliable. For example, consider a subnetwork manager that loses the connection to one of the managed elements it manages. In this case the alarm records relating to the complete object subtree starting at the object representing the managed element are not updated and more and hence unreliable.

Alarm lists advertise unreliable parts by indicating the base objects of unreliable subtrees in the (multi-valued) unreliable alarm scope attribute. When the complete alarm list is unreliable the unreliable alarm scope attribute shall specify the object instance of the MnS agent. When the bad part of the alarm list has been rebuilt and is up to date again the corresponding base object of the previously unreliable subtree is removed from the unreliable alarm scope attribute. An empty attribute indicates that the complete alarm list is reliable,

## 6.12 Alarm notifications

When objects are created or deleted, or when attribute values are updated, then this is normally notified to MnS consumers using object creation, object deletion or attribute value change notifications. When alarm records are created, or deleted or modified these general-purpose notifications are not used. Dedicated notifications are used instead as follows:

- If a new alarm record is added to an alarm list a notify new alarm notification is sent.

- If the acknowledgement state changes its value, the notify acknowledgment state changed notification is sent.

- If a comment is added to an alarm record, the notify comments notification is sent.

- If the correlated notifications attribute or the root cause indicator attribute changes its value, the notify correlated notification changed notification is sent.

- If the perceived severity changes its value to cleared, the notify cleared alarm notification is sent.

- In all other cases a notify changed alarm general notification is sent.

Alarms are identified in alarm notifications using the alarm identifier, except for in the notify new alarm notification, where the four alarm identifying attributes are included as well to allow the MnS consumer receiving the notification to relate the alarm identifier to the alarm identifying attributes.

When the perceived severity attribute changes its value, except to cleared, the MnS producer may send instead of a notify changed alarm general notification a notify changed alarm notification.

The removal of an alarm record from an alarm list is not notified directly, only indirectly through the notifications reporting the clearance and, if supported, the acknowledgement of an alarm:

- If alarm acknowledgement is not supported, the MnS consumer can deduct from the reception of a notification reporting the clearance of an alarm that the corresponding alarm record was removed from the alarm list.

- If alarm acknowledgement is supported, the MnS consumer can deduct from the consecutive reception of a notification reporting the clearance of an alarm and a notification reporting the acknowledgement of the same alarm that the corresponding alarm record was removed from the alarm list. The order of receiving the notifications is not relevant.

A MnS producer can maintain an exact copy of the alarm list on the MnS producer by consuming the alarm notifications, assuming of course the MnS consumer starts with an exact alarm list copy.

Modifications of the unreliable alarm scope attribute are notified using the notify potential faulty alarm list notification and the notify alarm list rebuilt notification. More specifically, when

- a new value is added to the unreliable alarm scope attribute the notify potential faulty alarm list notification is sent. The object class and object instance parameters of the notification header specify the base object of the subtree that has become unreliable.

- a value is removed from the unreliable alarm scope attribute the notify alarm list rebuilt notification is sent. The object class and object instance parameters of the notification header specify the base object of the subtree that has been rebuilt and is reliable again.

When (parts of) the alarm list are unreliable the MnS producer may nevertheless send reliable alarm notifications that allow a MnS consumer to maintain an exact copy of the (unreliable) alarm list on the MnS producer. When the MnS consumer receives an alarm list rebuilt notification he knows that his alarm list copy is reliable and no alignment with the alarm list on the MnS consumer is required. To inform the MnS consumer about if unreliable or reliable alarm notifications were sent, or in other words, if an alarm list alignment is required or not required the alarm list alignment required attribute is provided.

To receive the notifications described in this clause, MnS consumers need to have appropriate notification subscriptions in place.

## 6.13 Alarm list states

The alarm list features the operational state and the administrative state attribute.

When an alarm list is unlocked and enabled alarm records shall be added, updated, or removed based on currently prevailing alarm conditions. The alarm list is always representing the current alarm conditions. Alarm notifications are sent.

When an alarm list is locked, the system shall not add, delete, or update alarm records. However, the MnS consumer may acknowledge, clear or comment alarms. Alarm notifications are not sent.

When the alarm list is disabled, its behaviour is undefined, however the administrative state and operational state shall be correctly handled. Alarm records may or may not be added, deleted, or updated based on prevailing alarm conditions. Furthermore, the result of a MnS consumer acknowledging, clearing, or commenting an alarm is not predictable and may or may not fail. Alarm notifications are not sent.

When an alarm list is locked or disabled its alarm records are hence not reliable.

The operational state and administrative state attributes always represent the current state, and attribute value change notifications for these state attributes are always sent, even when the alarm list is locked or disabled.

Note that when moving from a locked or disabled state to an unlocked and enabled state it may take some time until all alarm records are updated, and the alarm list represents the current state of the system. The alarm list may be unreliable even though unlocked and enabled.

The system may advertise that the alarm list is unreliable in its entirety by setting the value of the unreliable alarm scope attribute to the Distinguished Name (DN) of the MnS agent.

## 6.14 Alarm record life cycle

When the system detects a fault, an error or failure caused by a fault, the system creates an internal alarm description based on the alarm record attributes. In a second step the system needs to determine if this internal alarm is a new alarm or just an update of an already existing alarm. It does so by checking if there is already an alarm record with the same values for the four alarm identifying attributes (object instance, alarm type, probable cause, and specific problem) in the alarm list.

- If there is an alarm record with the same values for the alarm identifying attributes, then the corresponding existing alarm record in the alarm list is updated.

- If there is no alarm record with the same values for the alarm identifying attributes, then a new alarm record is added to the alarm list.

If alarm acknowledgement is supported, alarm records for cleared alarms are deleted by the system only when they are acknowledged. In other words, the alarm list contains only alarm records for alarms, whose:

- perceived severity is not cleared, or whose

- perceived severity is cleared, but that are not acknowledged.

If alarm acknowledgement is not supported, alarm records for cleared alarms are deleted immediately by the system.

The alarms represented by the alarm records in the alarm list are also referred to as active alarms.

# 7 Model

## 7.1 Imported information entities and local labels

|  |  |
| --- | --- |
| Label reference | **Local label** |
| 3GPP 1 28.532 [2], notification, notifyMOICreation | notifyMOICreation |
| 3GPP 1 28.532 [2], notification, notifyMOIDeletion | notifyMOIDeletion |
| 3GPP 1 28.532 [2], notification, notifyMOIAttributeValueChanges | notifyMOIAttributeValueChanges |
| 3GPP 1 28.532 [2], notification, notifyMOIChanges | notifyMOIChanges |
| 3GPP 1 28.622 [5], IOC, Top | Top |
| 3GPP 1 28.622 [5], IOC, ManagedElement | ManagedElement |
| 3GPP 1 28.622 [5], IOC, SubNetwork | SubNetwork |
| 3GPP 1 28.622 [5], IOC, NtfSubscriptionControl | NtfSubscriptionControl |
| 3GPP 1 28.622 [5], IOC, HeartbeatControl | HeartbeatControl |
| 3GPP 1 28.622 [5], data type, ThresholdInfo | ThresholdInfo |

## 7.2 Class diagrams

### 7.2.1 Relationships

This clause depicts the set of classes (e.g. IOCs) implemented by Fault Management. This clause provides the overview of the relationships of relevant classes in UML. Subsequent clauses provide more detailed specification of various aspects of these classes.

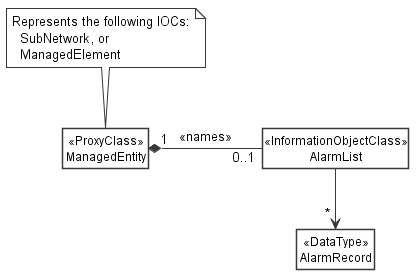


Figure 7.2.1-1: FM control NRM fragment

### 7.2.2 Inheritance

This clause depicts the inheritance relationships.

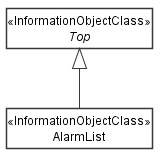


Figure 7.2.2-1: FM control NRM fragment

## 7.3 Class definitions

### 7.3.1 AlarmRecord <<dataType>>

#### 7.3.1.1 Definition

An AlarmRecord contains alarm information of an alarmed object instance. A new record is created in the alarm list when an alarmed object instance generates an alarm and no alarm record exists with the same values for objectInstance, alarmType, probableCause and specificProblem. When a new record is created the MnS producer creates an alarmId, that unambiguously identifies an alarm record in the AlarmList.

Alarm records are maintained only for active alarms. Inactive alarms are automatically deleted by the MnS producer from the AlarmList. Active alarms are alarms whose

a) perceivedSeverity is not "CLEARED", or whose

b) perceivedSeverity is "CLEARED" and its ackState is not "ACKNOWLEDGED" and alarm acknowledgement by the consumer is supported.

#### 7.3.1.2 Attributes

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Attribute name** | **S** | **isReadable** | **isWritable** | **isInvariant** | **isNotifyable** |
| alarmId | M | T | F | T | F |
| objectInstance | M | T | F | T | F |
| notificationId | M | T | F | F | F |
| alarmRaisedTime | M | T | F | F | F |
| alarmChangedTime | O | T | F | F | F |
| alarmClearedTime | M | T | F | F | F |
| alarmType | M | T | F | T | F |
| probableCause | M | T | F | T | F |
| specificProblem | O | T | F | T | F |
| perceivedSeverity | M | T | T (note) | F | F |
| backedUpStatus | O | T | F | F | F |
| backUpObject | O | T | F | F | F |
| trendIndication | O | T | F | F | F |
| thresholdInfo | O | T | F | F | F |
| stateChangeDefinition | O | T | F | F | F |
| monitoredAttributes | O | T | F | F | F |
| proposedRepairActions | O | T | F | F | F |
| additionalText | O | T | F | F | F |
| additionalInformation | O | T | F | F | F |
| rootCauseIndicator | CO | T | F | F | F |
| correlatedNotifications | CO | T | F | F | F |
| comments | O | T | T | F | F |
| ackTime | CM | T | F | F | F |
| ackUserId | CM | T | T | F | F |
| ackSystemId | CO | T | T | F | F |
| ackState | CM | T | T | F | F |
| clearUserId | CM | T | T | F | F |
| clearSystemId | CM | T | T | F | F |
| serviceUser | CM | T | F | F | F |
| serviceProvider | CM | T | F | F | F |
| securityAlarmDetector | CM | T | F | F | F |
| NOTE: This isWritable property is True only if alarm clearing by MnS consumers is supported . | | | | | |

#### 7.3.1.3 Attribute constraints

|  |  |  |
| --- | --- | --- |
| **Name** | | **Definition** |
| rootCauseIndicator  correlatedNotifications | At least one of these attributes shall be supported if the MnS producer supports alarm correlation. | |
| comments | This attribute shall be supported if the MnS producer supports alarm commenting | |
| ackTime  ackUserId  ackState  ackSystemId | These attributes should be supported if the MnS producer supports the alarm acknowledgement feature. | |
| clearUserId  clearSystemId | | These attributes shall be supported for alarm records that represent ADMC alarms. |
| serviceUser  serviceProvider  securityAlarmDetector | These attributes shall be supported for alarm records that represent security alarms. | |

#### 7.3.1.4 Notifications

See clause 7.5.

### 7.3.2 AlarmList

#### 7.3.2.1 Definition

The AlarmList represents the capability to store and manage alarm records. It can be name-contained by SubNetwork or ManagedElement. The management scope of an AlarmList is defined by all descendant objects of the base managed object, which is the object name-containing the AlarmList, and the base object itself. *AlarmList* MOIs should not be contained by a ManagedElement MOI if the ManagedElement MOI is contained in a Subnetwork that also contains an *AlarmList* MOI: multiple *AlarmList* MOIs with overlapping scopes should be avoided. In case an AlarmList is created under a ManagedElement that is also contained under a SubNetwork which also has an AlarmList child MOI, alarms in scope of that ManagedElement shall only be handled by the ManagedElement's AlarmList and shall not be visible in the SubNetwork's AlarmList.

AlarmList instance(s) are created by the system or are pre-installed. They cannot be created nor deleted by MnS consumers.

An instance of SubNetwork or ManagedElement has at most one name-contained instance of AlarmList.

When the alarm list is locked or disabled, its attributes (except the administrativeState/operationalState) may contain any unreliable data. No alarm notifications are sent by the MnS producer.

#### 7.3.2.2 Attributes

The AlarmList IOC includes attributes inherited from Top IOC and the following attributes:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Attribute Name** | **S** | **isReadable** | **isWritable** | **isInvariant** | **isNotifyable** |
| administrativeState | O | T | T | F | T |
| operationalState | M | T | F | F | T |
| numOfAlarmRecords | M | T | F | F | F |
| lastModification | M | T | F | F | F |
| alarmRecords | M | T | T | F | F |
| unreliableAlarmScope | O | T | F | F | F |

#### 7.3.2.3 Attribute constraints

None.

#### 7.3.2.4 Notifications

The common notifications defined in clause 7.5 are valid for this IOC, without exceptions or additions.

### 7.3.3 AlarmComment <<dataType>>

#### 7.3.3.1 Definition

This data type represents a comment on an alarm.

#### 7.3.3.2 Attributes

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Attribute Name** | **S** | **isReadable** | **isWritable** | **isInvariant** | **isNotifyable** |
| commentTime | M | T | F | T | F |
| commentUserId | M | T | T | T | F |
| commentSystemId | O | T | T | T | F |
| commentText | M | T | T | T | F |

#### 7.3.3.3 Attribute constraints

None

#### 7.3.3.4 Notifications

See clause 7.5.

### 7.3.4 CorrelatedNotification <<dataType>>

#### 7.3.4.1 Definition

The sourceObjectInstance attribute of CorrelatedNotification identifies one MonitoredEntity. For the MonitoredEntity identified, a set of notification identifiers is also identified. One or more CorrelatedNotification instances can be included in an AlarmRecord. In this case, the information of the AlarmRecord is said to be correlated to information carried in the notifications identified by the CorrelatedNotification instances. See further definition of correlated notification in ITU-T Recommendation X.733 [8], clause 8.1.2.9.

The notification identified by the CorrelatedNotification, as defined in ITU-T and used here, can carry all types of information and is not restricted to carrying alarm information only. For example, a notification, identified by the CorrelatedNotification, can indicate a managed instance attribute value change. In this case, the information of the AlarmRecord is said to be correlated to the managed instance attribute value change event.

If a CorrelatedNotification references an alarm (e.g., by referencing the notificationId of a notifyNewAlarm notification), the alarmRecord for that alarm may or may not exist in the AlarmList. For example, the alarm may have been acknowledged and cleared and therefore, removed from the AlarmList.

#### 7.3.4.2 Attributes

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Attribute Name** | **S** | **isReadable** | **isWritable** | **isInvariant** | **isNotifyable** |
| sourceObjectInstance | M | T | F | F | F |
| notificationIdSet | M | T | F | F | F |

#### 7.3.4.3 Attribute constraints

None.

#### 7.3.4.4 Notifications

See clause 7.5.

## 7.4 Attribute definitions

### 7.4.1 Attribute properties

The following table defines the properties of attributes specified in the present document.

| **Attribute Name** | **Documentation and Allowed Values** | **Properties** |
| --- | --- | --- |
| objectClass | Class of a managed object instance. | type: String  multiplicity: 1  isOrdered: N/A  isUnique: N/A  defaultValue: None  isNullable: False |
| objectInstance | Managed object instance identified by its DN. | type: DN  multiplicity: 1  isOrdered: N/A  isUnique: N/A  defaultValue: None  isNullable: False |
| systemDN | Distinguished Name (DN) of an MnSAgent. | type: DN  multiplicity: 0..1  isOrdered: N/A  isUnique: N/A  defaultValue: None  isNullable: False |
| administrativeState | Administrative state of a managed object instance. The administrative state describes the permission to use or prohibition against using the object instance. The administrative state is set by the MnS consumer.  allowedValues: LOCKED, UNLOCKED. | type: ENUM  multiplicity: 1  isOrdered: N/A  isUnique: N/A  defaultValue: LOCKED  isNullable: False |
| operationalState | Operational state of manged object instance. The operational state describes if an object instance is operable ("ENABLED") or inoperable ("DISABLED"). This state is set by the object instance or the MnS producer and is hence READ-ONLY.  allowedValues: ENABLED, DISABLED. | type: ENUM  multiplicity: 1  isOrdered: N/A  isUnique: N/A  defaultValue: DISABLED  isNullable: False |
| alarmRecords | List of alarm records | type: AlarmRecord  multiplicity: \*  isOrdered: False  isUnique: True  default value: None  isNullable: False |
| numOfAlarmRecords | Number of alarm records in the AlarmList.  allowedValues: Non-negative numbers. | type: integer  multiplicity: 1  isOrdered: N/A  isUnique: N/A  defaultValue: None  isNullable: False |
| lastModification | Time an alarm record was modified the last time. | type: DateTime  multiplicity: 1  isOrdered: N/A  isUnique: N/A  defaultValue: None  isNullable: False |
| unreliableAlarmScope | Identifies, the part of the alarm scope that may not be reliable.  If this parameter is equal to the instance carried in systemDN, then all AlarmRecord instances in the AlarmList may not be reliable.    If this parameter is equal to some instance represented by MonitoredEntity, then only AlarmRecord related to this instance and its descendants may not be reliable. | type: DN  multiplicity: 0..\*  isOrdered: False  isUnique: True  defaultValue: None  isNullable: False |
| alarmId | Identifies an AlarmRecord in the AlarmList. The value is unique within the AlarmList MOI. | type: string  multiplicity: 1  isOrdered: N/A  isUnique: N/A defaultValue: None  isNullable: False |
| notificationId | The Id of the last notification sent as a consequence of updating the AlarmRecord. | type: integer  multiplicity: 1  isOrdered: N/A  isUnique: N/A defaultValue: None  isNullable: False |
| alarmRaisedTime | Date and time the alarm was raised. | type: DateTime  multiplicity: 1  isOrdered: N/A  isUnique: N/A defaultValue: None  isNullable: False |
| alarmChangedTime | It indicates the last date and time when the AlarmInformation is changed by the alarmed resource. Changes to AlarmInformation caused by invocations of the management service consumer would not change this date and time. | type: DateTime  multiplicity: 0..1  isOrdered: N/A  isUnique: N/A defaultValue: None  isNullable: False |
| alarmClearedTime | Date and time the alarm was cleared. | type: DateTime  multiplicity: 0..1  isOrdered: N/A  isUnique: N/A defaultValue: None  isNullable: False |
| alarmType | It indicates the type of alarm.  Communications Alarm:  An alarm of this type is associated with the procedure and/or process required conveying information from one point to another (ITU-T Recommendation X.733 [8]).  Quality of Service Alarm:  An alarm of this type is associated with degradation in the quality of a service (ITU T Recommendation X.733 [8]).  Processing Error Alarm:  An alarm of this type is associated with a software or processing fault (ITU T Recommendation X.733 [8]).  Equipment Alarm:  An alarm of this type is associated with an equipment fault (ITU-T Recommendation X.733 [8]).  Environmental Alarm:  An alarm of this type is associated with a condition related to an enclosure in which the equipment resides (ITU-T Recommendation X.733 [8]).  Security related alarm types  Integrity Violation:  An indication that information may have been illegally modified, inserted or deleted.  Operational Violation:  An indication that the provision of the requested service was not possible due to the unavailability, malfunction or incorrect invocation of the service.  Physical Violation:  An indication that a physical resource has been violated in a way that suggests a security attack.  Security Service or Mechanism Violation:  An indication that a security attack has been detected by a security service or mechanism.  Time Domain Violation: An indication that an event has occurred at an unexpected or prohibited time.  Allow values:  COMMUNICATIONS\_ALARM,  QUALITY\_OF\_SERVICE\_ALARM,  PROCESSING\_ERROR\_ALARM,  EQUIPMENT\_ALARM,  ENVIRONMENTAL\_ALARM,  INTEGRITY\_VIOLATION,  OPERATIONAL\_VIOLATION,  PHYSICAL\_VIOLATION, SECURITY\_SERVICE\_OR\_MECHANISM\_VIOLATION, TIME\_DOMAIN\_VIOLATION | type: ENUM  multiplicity: 1  isOrdered: N/A  isUnique: N/A defaultValue: None  isNullable: False |
| probableCause | It qualifies alarm and provides further information than alarmType. This attribute value shall be single-value and of simple type such as integer or string. See Annex A for a complete listing. | type: string or integer  multiplicity: 1  isOrdered: N/A  isUnique: N/A defaultValue: None  isNullable: False |
| specificProblem | It provides further refinement to the probableCause. This attribute value shall be single-valued and of simple type such as integer or string. See definition in ITU-T Recommendation X.733 [8] clause 8.1.2.2. | type: string or integer  multiplicity: 0..1  isOrdered: N/A  isUnique: N/A defaultValue: None  isNullable: False |
| perceivedSeverity | It indicates the relative level of urgency for operator attention.  allowedValues:  CRITICAL,  MAJOR,  MINOR,  WARNING,  INDETERMINATE,  CLEARED | type: ENUM  multiplicity: 1  isOrdered: N/A  isUnique: N/A defaultValue: None  isNullable: False |
| backedUpStatus | It indicates if an object (the MonitoredEntity) has a back up. See definition in ITU-T Recommendation X.733 [8] clause 8.1.2.4. | type: boolean  multiplicity: 0..1  isOrdered: N/A  isUnique: N/A defaultValue: False  isNullable: False |
| backUpObject | Backup object of the alarmed object as defined in ITU-T Rec. X. 733 [8] | type: DN  multiplicity: 0..1  isOrdered: N/A  isUnique: N/A defaultValue: None  isNullable: False |
| trendIndication | It indicates if some observed condition is getting better, worse, or not changing.  AllowedValues:  MORE\_SEVERE, NO\_CHANGE, LESS\_SEVERE | type: ENUM  multiplicity: 0..1  isOrdered: N/A  isUnique: N/A defaultValue: None  isNullable: False |
| thresholdInfo | It indicates the crossed threshold information such as:  - The identifier of the monitored attribute whose value has crossed a threshold,  - The threshold settings,  - The observed value that have crossed a threshold, etc.  See definition in ITU-T Recommendation X.733 [8] clause 8.1.2.7. See also for information in 1 32.401 [12] clause 5.6. | type: ThresholdInfo  multiplicity: \*  isOrdered: False  isUnique: True defaultValue: None  isNullable: False |
| stateChangeDefinition | It indicates attribute value changes associated with the alarm for state attributes of the monitored entity (state transitions). The change is reported with the name of the state attribute, the new value and an optional old value. See definition in ITU-T Recommendation X.733 [8] clause 8.1.2.11.  The content of the attribute is a list of attributeNames and attributeValues. AttributeValues may be complex types.  Beside the new value it may contain the old value as well. | type: AttributeValueChange  multiplicity: 0..\*  isOrdered: N/A  isUnique: N/A defaultValue: None  isNullable: False |
| monitoredAttributes | It indicates attributes of the monitored entity and their values at the time the alarm occurred that are of interest for the alarm report. How these attributes are chosen is outside of the scope of the present document. See definition in ITU-T Recommendation X.733 [8] clause 8.1.2.11.  The content of the attribute is a list of attributeName- attributeValue pairs. AttributeValues may be complex types. | type: NameValuePair  multiplicity: \*  isOrdered: False  isUnique: True defaultValue: None  isNullable: False |
| proposedRepairActions | Used if the cause is known and the system being managed can suggest one or more solutions to fix the problem causing the alarm as defined in ITU-T Recommendation X. 733 [8] | type: string  multiplicity: 0..1  isOrdered: N/A  isUnique: N/A defaultValue: None  isNullable: False |
| additionalText | Allows a free form text description to be reported as defined in ITU-T Recommendation X. 733 [8]. | type: string  multiplicity: 0..1  isOrdered: N/A  isUnique: N/A defaultValue: None  isNullable: False |
| additionalInformation | This attribute when present allows the inclusion of a set of vendor specific alarm information in the alarm.  A specific condition for this optional population is when an alarm presented by the Management System (e.g. via the user interface) has different values of perceived severity, and / or alarm type, compared with the values presented to the Itf-N.  Any other uses of additional information on the alarm and its semantics is outside the scope of the present document  The content of the attribute is a list of attributeNames and string attributeValues. | type: NameValuePair  multiplicity: \*  isOrdered: False  isUnique: True defaultValue: None  isNullable: False |
| rootCauseIndicator | It indicates that this AlarmInformation is the root cause of the events captured by the notifications whose identifiers are in the related CorrelatedNotification instances. | type: boolean  multiplicity: 0..1  isOrdered: N/A  isUnique: N/A defaultValue: None  isNullable: False |
| ackTime | It identifies the time when the alarm has been acknowledged or unacknowledged the last time, i.e. it registers the time when ackState changes. | type: DateTime  multiplicity: 0..1  isOrdered: N/A  isUnique: N/A defaultValue: None  isNullable: False |
| ackUserId | It identifies the last user who has changed the acknowledgement state. | type: string  multiplicity: 0..1  isOrdered: N/A  isUnique: N/A defaultValue: None  isNullable: False |
| ackSystemId | It identifies the system that last changed the ackState of an alarm, i.e. acknowledged or unacknowledged the alarm. | type: string  multiplicity: 0..1  isOrdered: N/A  isUnique: N/A defaultValue: None  isNullable: False |
| ackState | It identifies the acknowledgement state of an alarm.  AllowedValues: ACKNOWLEDGED, UNACKNOWLEDGED | type: ENUM  multiplicity: 1  isOrdered: N/A  isUnique: N/A defaultValue: None  isNullable: False |
| clearUserId | It carries the identity of the user who invokes the clearAlarms operation. | type: string  multiplicity: 0..1  isOrdered: N/A  isUnique: N/A defaultValue: None  isNullable: False |
| clearSystemId | Identifier of a system clearing an alarm | type: string  multiplicity: 0..1  isOrdered: N/A  isUnique: N/A defaultValue: None  isNullable: False |
| serviceUser | It identifies the service-user whose request for service provided by the serviceProvider led to the generation of the security alarm. | type: string  multiplicity: 0..1  isOrdered: N/A  isUnique: N/A defaultValue: None  isNullable: False |
| serviceProvider | It identifies the service-provider whose service is requested by the serviceUser and the service request provokes the generation of the security alarm. | type: string  multiplicity: 0..1  isOrdered: N/A  isUnique: N/A defaultValue: None  isNullable: False |
| securityAlarmDetector | It carries the identity of the detector of the security alarm. | type: string  multiplicity: 0..1  isOrdered: N/A  isUnique: N/A defaultValue: None  isNullable: False |
| comments | List of comments and data about the comments. | type: AlarmComment  multiplicity: \*  isOrdered: False  isUnique: True defaultValue: None  isNullable: False |
| correlatedNotifications | List of correlated notifications. | type: CorrelatedNotification  multiplicity: \*  isOrdered: False  isUnique: True defaultValue: None  isNullable: False |
| commentTime | Date and Time the comment was created. | type: DateTime  multiplicity: 1  isOrdered: N/A  isUnique: N/A defaultValue: None  isNullable: False |
| commentUserId | It carries the identification of the user who made the comment. | type: string  multiplicity: 1  isOrdered: N/A  isUnique: N/A defaultValue: None  isNullable: False |
| commentSystemId | It carries the identification of the system (Management System) from which the comment is made. That system supports the user that made the comment. | type: string  multiplicity: 1  isOrdered: N/A  isUnique: N/A defaultValue: None  isNullable: False |
| commentText | It carries the textual comment. | type: string  multiplicity: 1  isOrdered: N/A  isUnique: N/A defaultValue: None  isNullable: False |
| CorrelatedNotification.sourceObjectInstance | It identifies one MonitoredEntity. It is unique within a multivalue attribute based on the CorrelatedNotification data type. | type: DN  multiplicity: 1  isOrdered: N/A  isUnique: N/A defaultValue: None  isNullable: False |
| CorrelatedNotification.notificationIdSet | A list of correlated notificationIds. | type: integer  multiplicity: 1..\*  isOrdered: False  isUnique: True defaultValue: None  isNullable: False |
| NOTEs: none. | | |

### 7.4.2 Constraints

None.

## 7.5 Common notifications

### 7.5.1 Alarm notifications

This clause presents a list of notifications, defined in clause 9, that a MnS consumer can receive. The notification header attribute objectClass/objectInstance captures the DN of an instance of an IOC defined in the present document.

| **Name** | **Notes** |
| --- | --- |
| notifyNewAlarm |  |
| notifyClearedAlarm |  |
| notifyAlarmListRebuilt |  |
| notifyChangedAlarmGeneral |  |
| notifyChangedAlarm |  |
| notifyCorrelatedNotificationChanged |  |
| notifyAckStateChanged |  |
| notifyComments |  |
| notifyPotentialFaultyAlarmList |  |

### 7.5.2 Configuration notifications

This clause presents a list of notifications, defined in [2], that a MnS consumer can receive. The notification header attribute objectClass/objectInstance, captures the DN of an instance of an IOC defined in the present document.

| **Name** | **Notes** |
| --- | --- |
| notifyMOICreation |  |
| notifyMOIDeletion |  |
| notifyMOIAttributeValueChanges |  |
| notifyMOIChanges |  |

# 8 Notifications

## 8.1 Overview

This clause specifies the alarm notifications used to report modifications of the alarm list and alarm records. To receive these notifications MnS consumers must have appropriate subscriptions in place. TS 28.622 [5], clause 4.3.22 describes how to manage notification subscriptions.

## 8.2 notifyNewAlarm

### 8.2.1 Definition

This notification is generated by the MnS producer when a new alarm is raised and an *AlarmRecord* is added to the AlarmList. The notification parameters depend on the alarmType and are different for non-security and security alarms.

### 8.2.2 Input parameters

If the alarmType is "Communications Alarm", "Processing Error Alarm", "Environmental Alarm". "Quality Of Service Alarm" or "Equipment Alarm" the alarm is considered to be non-security related. If the alarmType is "Integrity Violation", "Operational Violation", "Physical Violation", "Security Service or Mechanism Violation" or "Time Domain Violation" the alarm is considered to be security related.

Table 8.2.2-1: Input parameters for notifyNewAlarm

| **Parameter Name** | **S** | **Matching Information/ Information Type / Legal Values** | **Description** |
| --- | --- | --- | --- |
| objectClass | M | String  ClassName of the object identified by objectInstance. |  |
| objectInstance | M | alarmRecord.objectInstance  DN of the MonitoredEntity that is the source of the alarm |  |
| notificationId | M | This is an identifier for the notification, which may be used to correlate notifications. | The identifier of the notification shall be chosen to be unique across all notifications of a particular managed object instance throughout the time that correlation is significant, it uniquely identifies the notification from other notifications generated by the subject MOI. |
| notificationType | M | "notifyNewAlarm" |  |
| eventTime | M | alarmRecord.alarmRaisedTime |  |
| systemDN | M | It shall carry the DN of management service providers; the DN of an MnsAgent MOI [5]. |  |
| alarmId | M | alarmRecord.alarmId |  |
| alarmType | M | alarmRecord.alarmType |  |
| probableCause | M | alarmRecord.probableCause |  |
| perceivedSeverity | M | alarmRecord.perceivedSeverity |  |
| specificProblem | O | alarmRecord.specificProblem |  |
| backedUpStatus | CO | alarmRecord.backedUpStatus | Used only in non-security notifications. |
| backUpObject | CO | alarmRecord.backUpObject | Used only in non-security notifications. |
| trendIndication | CO | alarmRecord.trendIndication | Used only in non-security notifications. |
| thresholdInfo | CO | alarmRecord.thresholdInfo | Used only in non-security notifications. |
| correlatedNotifications | O | alarmRecord.correlatedNotifications |  |
| stateChangeDefinition | CO | alarmRecord.stateChangeDefinition | Used only in non-security notifications. |
| monitoredAttributes | CO | alarmRecord.monitoredAttributes | Used only in non-security notifications. |
| proposedRepairActions | CO | alarmRecord.proposedRepairActions | Used only in non-security notifications. |
| additionalText | O | alarmRecord.additionalText |  |
| additionalInformation | O | alarmRecord.additionalInformation |  |
| rootCauseIndicator | O | alarmRecord.rootCauseIndicator |  |
| serviceUser | CM | alarmRecord.securityServiceUser | Used only in security notifications.  This may contain no information if the identify of the service-user (requesting the service) is not known. |
| serviceProvider | CM | alarmRecord.securityServiceProvider | Used only in security notifications.  This shall always identify the service-provider receiving a service request, from serviceUser, that provokes the security alarm. |
| securityAlarmDetector | CM | alarmRecord.securityAlarmDetector | Used only in security notifications.  This may contain no information if the detector of the security alarm is the serviceProvider. |

## 8.3 notifyClearedAlarm

### 8.3.1 Definition

This notification is generated by the MnS producer when the perceivedSeverity of an existing AlarmRecord changes to "CLEARED"; the AlarmRecord may be removed when sending the notification.

### 8.3.2 Input parameters

Table 8.3.2-1: Input parameters for notifyClearedAlarm

| **Parameter Name** | **S** | **Matching Information/ Information Type / Legal Values** | **Comment** |
| --- | --- | --- | --- |
| objectClass | M | See clause 8.2.1 |  |
| objectInstance | M | alarmRecord.objectInstance  DN of the MonitoredEntity that is the source of the alarm |  |
| notificationId | M | See clause 8.2.1 |  |
| notificationType | M | "notifyClearedAlarm" |  |
| eventTime | M | alarmRecord.alarmClearedTime |  |
| systemDN | M | See clause 8.2.1 |  |
| alarmId | M | alarmRecord.alarmId |  |
| alarmType | M | alarmRecord.alarmType |  |
| probableCause | M | alarmRecord.probablaCause |  |
| perceivedSeverity | M | alarmRecord.perceivedSeverity | Value shall be "CLEARED" |
| correlatedNotifications | O | alarmRecord.correlatedNotifications |  |
| clearUserId | O | alarmRecord.clearUserId | This parameter shall be present if the AlarmRecord is cleared by the consumer. |
| clearSystemId | O | alarmRecord.clearSystemId | This parameter shall be present if clearUserId is present |

## 8.4 notifyChangedAlarmGeneral

### 8.4.1 Definition

This notification is generated by the MnS producer when one or more of the following attributes of an AlarmRecord instance in the AlarmList changes its value: perceivedSeverity, backedUpStatus, backUpObject, trendIndication, thresholdInfo, stateChangeDefinition, monitoredAttributes, proposedRepairActions, additionalText, additionalInformation, serviceUser, serviceProvider or securityAlarmDetector. From the attributes listed above, only those that changed value shall be included in the notification.

The notification parameters depend on the alarmType and are different for non-security and security alarms. If the alarmType is "Communications Alarm", "Processing Error Alarm", "Environmental Alarm". "Quality Of Service Alarm" or "Equipment Alarm" the alarm is considered to be non-security related. If the alarmType is "Integrity Violation", "Operational Violation", "Physical Violation", "Security Service or Mechanism Violation" or "Time Domain Violation" the alarm is considered to be security related.

### 8.4.2 Input parameters

Table 8.4.2-1: Input parameters for notifyChangedAlarmGeneral

| **Parameter Name** | **S** | **Matching Information/ Information Type / Legal Values** | **Comment** |
| --- | --- | --- | --- |
| objectClass | M | See clause 8.2.1 |  |
| objectInstance | M | alarmRecord.objectInstance  DN of the MonitoredEntity that is the source of the alarm |  |
| notificationId | M | See clause 8.2.1 |  |
| notificationType | M | "notifyChangedAlarmGeneral" |  |
| eventTime | M | alarmRecord.alarmChangedTime |  |
| systemDN | M | See clause 8.2.1 |  |
| alarmId | M | alarmRecord.alarmId |  |
| alarmType | M | alarmRecord.alarmType |  |
| probableCause | M | alarmRecord.probableCause |  |
| specificProblem | O | alarmRecord.specificProblem |  |
| perceivedSeverity | O | alarmRecord.perceivedSeverity |  |
| backedUpStatus | CO | alarmRecord.backedUpStatus | Used only in non-security notifications. |
| backUpObject | CO | alarmRecord.backUpObject | Used only in non-security notifications. |
| trendIndication | CO | alarmRecord.trendIndication | Used only in non-security notifications. |
| thresholdInfo | CO | alarmRecord.thresholdInfo | Used only in non-security notifications. |
| correlatedNotifications | O | alarmRecord.correlatedNotifications |  |
| stateChangeDefinition | CO | alarmRecord.stateChange | Used only in non-security notifications. |
| monitoredAttributes | CO | alarmRecord.monitoredAttributes | Used only in non-security notifications. |
| proposedRepairActions | CO | alarmRecord.proposedRepairActions | Used only in non-security notifications. |
| additionalText | O | alarmRecord.additionalText |  |
| additionalInformation | O | alarmRecord.additionalInformation |  |
| rootCauseIndicator | O | alarmRecord.rootCauseIndicator |  |
| serviceUser | C | alarmRecord.securityServiceUser | Used only in security notifications.  This may contain no information if the identify of the service-user (requesting the service) is not known. |
| serviceProvider | C | alarmRecord.securityServiceProvider | Used only in security notifications.  This shall always identify the service-provider receiving a service request, from serviceUser, that provokes the security alarm. |
| securityAlarmDetector | C | alarmRecord.securityAlarmDetector | Used only in security notifications.  This may contain no information if the detector of the security alarm is the serviceProvider. |
| changedAlarmAttributes | O | LIST OF SEQUENCE <AttributeName, OldAttributeValue> | The changed alarm attributes (name/value pairs) (with old values). |

## 8.5 notifyAlarmListRebuilt

### 8.5.1 Definition

This notification is generated by the MnS producer when the AlarmList has been completely or partially rebuilt.

### 8.5.2 Input parameters

Table 8.5.2-1: Input parameters for notifyAlarmListRebuilt

| **Parameter Name** | **S** | **Matching Information/ Information Type / Legal Values** | **Comment** |
| --- | --- | --- | --- |
| objectClass | M | See clause 8.2.1 |  |
| objectInstance | M | DN | Identifies the part of the alarm scope that has been rebuilt.  If this parameter is equal to the instance carried in systemDN, then all AlarmRecord instances in the AlarmList may have been rebuilt.    If this parameter is equal to some other instance, then only alarmRecords related to this instance and its descendants may have been rebuilt. |
| notificationId | M | See clause 8.2.1. |  |
| notificationType | M | "notifyAlarmListRebuilt" |  |
| eventTime | M | DateTime | The time when the alarm list rebuilt process was completed. |
| systemDN | M | See clause 8.2.1 |  |
| reason | M | String  "System-NE communication error", "System restarts", "indeterminate". Other values can be added. | The reason why the system has rebuilt the AlarmList. This may carry different reasons than that carried by the immediate previous notifyPotentialFaultyAlarmList. |
| alarmListAlignmentRequirement | O | "alignmentRequired", "alignmentNotRequired". | Indicates whether the AlarmList consumer should re-read the AlarmList. This is needed if the producer has failed to send some notifications needed for the consumer to follow the content and changes in the AlarmList. |

## 8.6 notifyChangedAlarm

### 8.6.1 Definition

This notification is generated by the MnS producer when the perceivedSeverity of an existing AlarmRecord changes (except to the value "CLEARED").

The notification is **deprecated**, use notifyChangedAlarmGeneral instead.

### 8.6.2 Input parameters

Table 8.6.2-1: Input parameters for notifyChangedAlarm

| **Parameter Name** | **S** | **Matching Information/ Information Type / Legal Values** | **Comment** |
| --- | --- | --- | --- |
| objectClass | M | See clause 8.2.1 |  |
| objectInstance | M | alarmRecord.objectInstance  DN of the MonitoredEntity that is the source of the alarm |  |
| notificationId | M | See clause 8.2.1 |  |
| notificationType | M | "notifyChangedAlarm" |  |
| eventTime | M | alarmRecord.alarmChangedTime |  |
| systemDN | M | See clause 8.2.1 |  |
| alarmId | M | alarmRecord.alarmId |  |
| alarmType | M | alarmRecord.alarmType |  |
| probableCause | M | alarmRecord.probableCause |  |
| perceivedSeverity | M | alarmRecord.perceivedSeverity |  |

## 8.7 notifyCorrelatedNotificationChanged

### 8.7.1 Definition

This notification is generated by the MnS producer when the set of correlatedNotifications is created, updated or deleted.

### 8.7.2 Input parameters

Table 8.7.2-1: Input parameters for notifyCorrelatedNotificationChanged

| **Parameter Name** | **S** | **Matching Information/ Information Type / Legal Values** | **Comment** |
| --- | --- | --- | --- |
| objectClass | M | See clause 8.2.1 |  |
| objectInstance | M | alarmRecord.objectInstance  DN of the MonitoredEntity that is the source of the alarm |  |
| notificationId | M | See clause 8.2.1. |  |
| notificationType | M | "notifyCorrelatedNotificationChanged" |  |
| eventTime | M | alarmRecord.alarmChanedTime  It carries the time when the CorrelatedNotification is created, updated or deleted. |  |
| systemDN | M | See clause 8.2.1 |  |
| alarmId | M | alarmRecord.alarmId |  |
| correlatedNotifications | M | alarmRecord.correlatedNotifications |  |
| rootCauseIndicator | O | alarmRecord.rootCauseIndicator |  |

## 8.8 notifyAckStateChanged

### 8.8.1 Definition

This notification is generated by the MnS producer when the acknowledgement state of an alarm changes from "UNACKNOWLEDGED" to "ACKNOWLEDGED" or back from "ACKNOWLEDGED" to "UNACKNOWLEDGED".

### 8.8.2 Input parameters

Table 8.8.2-1: Input parameters for notifyAckStateChanged

| **Parameter Name** | **S** | **Matching Information/ Information Type / Legal Values** | **Comment** |
| --- | --- | --- | --- |
| objectClass | M | See clause 8.2.1 |  |
| objectInstance | M | alarmRecord.objectInstance  DN of the MonitoredEntity that is the source of the alarm |  |
| notificationId | M | See clause 8.2.1 |  |
| notificationType | M | "notifyAckStateChanged" |  |
| eventTime | M | alarmRecord.ackTime |  |
| systemDN | M | See clause 8.2.1 |  |
| alarmId | M | alarmRecord.alarmId |  |
| alarmType | M | alarmRecord.alarmType |  |
| probableCause | M | alarmRecord.probableCause |  |
| perceivedSeverity | M | alarmRecord.perceivedSeverity |  |
| ackState | M | alarmRecord.ackState |  |
| ackUserId | M | alarmRecord.ackUserId |  |
| ackSystemId | O | alarmRecord.ackSystemId |  |

## 8.9 notifyComments

### 8.9.1 Definition

This notification is generated by the MnS producer when a Comment instance is updated in an AlarmRecord instance in the AlarmList.

### 8.9.2 Input parameters

Table 8.9.2-1: Input parameters for notifyComments

| **Parameter Name** | **S** | **Matching Information/ Information Type / Legal Values** | **Comment** |
| --- | --- | --- | --- |
| objectClass | M | See clause 8.2.1 |  |
| objectInstance | M | alarmRecord.objectInstance  DN of the MonitoredEntity that is the source of the alarm |  |
| notificationId | M | See clause 8.2.1 |  |
| notificationType | M | "notifyComments" |  |
| eventTime | M | alarmRecord.alarmChangedTime | The time the comment was updated |
| systemDN | M | See clause 8.2.1 |  |
| alarmId | M | alarmRecord.alarmId |  |
| alarmType | M | alarmRecord.alarmType |  |
| probableCause | M | alarmRecord.probableCause |  |
| perceived Severity | M | alarmRecord.perceivedSeverity |  |
| comments | M | The Comment instances related to this AlarmRecord.  Type: AlarmComment |  |

## 8.10 notifyPotentialFaultyAlarmList

### 8.10.1 Definition

This notification is generated by the MnS producer when the MnS producer looses confidence in the integrity of its alarm list.

The MnS producer may then rebuild the faulty alarm list. When the alarm List is rebuilt or confidence in the existing alarm list is re-established the MnS producer should generate a notifyAlarmListRebuilt notification.

The parameters objectClass and objectInstance are used to specify if the complete alarm list is unreliable or only parts thereof.

### 8.10.2 Input parameters

Table 8.10.2-1: Input parameters for notifyPotentialFaultyAlarmList

| **Parameter Name** | **S** | **Matching Information/ Information Type / Legal Values** | **Comment** |
| --- | --- | --- | --- |
| objectClass | M | See clause 8.2.1 |  |
| objectInstance | M | It identifies the instance identified by systemDN or an instance of MonitoredEntity. | Identifies, together with the objectClass parameter, the part of the alarm scope that may be unreliable.    If this parameter is equal to the instance carried in systemDN, then all AlarmRecord instances in the AlarmList may be unreliable.    If this parameter is equal to some other instance, then only AlarmRecords related to this instance and its descendants may be unreliable. |
| notificationId | M | T See clause 8.2.1. |  |
| notificationType | M | "notifyPotentialFaultyAlarmList" |  |
| eventTime | M | DateTime | Time when the MnS producer lost confidence in the integrity of the alarm list |
| systemDN | M | See clause 8.2.1 |  |
| reason | M | "serviceprovider-NE communication error", " serviceprovider restarts", "indeterminate". Other values can be added. | Reason why the MnS producer has to rebuild its AlarmList. |

Annex A (normative):   
Solution sets

## A.1 RESTful HTTP-based solution set

### A.1.1 Mapping of the NRM

The mapping of object classes and attributes follows the general rules defined in TS 32.160 [15], clause 10.3.

### A.1.2 Mapping of notifications

Principles:

- Only information not documented in the OpenAPI files is included in this clause.

- The following items are documented in the OpenAPI files: HTTP-Method, parameter name and type.

- The name of the parameter is the same in the stage 2 information model (clauses 8 and 9) and in the stage 3 OpenAPI definition. Exceptions, if any, are listed below.

Table A.1.2-1: Mapping of IS notification input parameters to SS equivalents (HTTP POST)

|  |  |  |  |
| --- | --- | --- | --- |
| **IS parameter name** | **SS parameter location** | **SS parameter name** | **SS parameter type** |
| objectClass | request body | href | Uri (see [10]) |
| objectInstance |

### A.1.3 OpenAPI definitions

OpenAPI definitions for the NRM are specified in Forge [11].

Directory: OpenAPI

Files:

TS28111\_FaultNrm.yaml

TS28111\_FaultNotifications.yaml

### A.1.4 Examples

**Retrieving alarms**

The first example shows how to retrieve an alarm based on its "alarmId".

|  |
| --- |
| GET /SubNetwork=SN1/AlarmList=AL1?\  fields=/attributes/alarmRecords/alarmId1 HTTP/1.1 |

Multiple alarms can be retrieved with the following request.

|  |
| --- |
| GET /SubNetwork=SN1/AlarmList=AL1?\  fields=/attributes/alarmRecords/(alarmId1, alarmId2) HTTP/1.1 |

The next example shows how all alarms with a perceived severity of major or critical can be retrieved.

|  |
| --- |
| GET /SubNetwork=SN1/AlarmList=AL1?\  selection=/AlarmList[id="AL1"]/attributes/alarmRecords\  /\*[perceivedSeverity="MAJOR" or perceivedSeverity="CRITICAL"] HTTP/1.1 |

To retrieve all alarms for a specific managed object instance identified by "DN1" the MnS consumer may send the following request.

|  |
| --- |
| GET /SubNetwork=SN1/AlarmList=AL1?\  selection=/AlarmList[id="AL1"]/attributes/alarmRecords\  /\*[objectInstance="DN1"] HTTP/1.1 |

A MnS consumer wants to retrieve often all alarms from one Managed Element. A Manged Element is modelled in the management system by an object tree whose base object is a "ManagedElement" instance. In the example below this instance is identified by the DN "example.com/SubNetwork=SN1/ManagedElement=ME1". The Jex expression in the query parameter "selection" evaluates to true for all DNs, that contain (start) with this DN, i.e. for all objects in the object subtree of interest.

|  |
| --- |
| GET /SubNetwork=SN1/AlarmList=AL1?\  selection=/AlarmList[id="AL1"]/attributes/alarmRecords\  /\*[contains(objectInstance,"example.com/SubNetwork=SN1/ManagedElement=ME1")] |

**Acknowledging alarms**

To acknowledge an alarm a MnS consumer has multiple alternatives. With JSON Patch the request may look as follows.

|  |
| --- |
| PATCH /SubNetwork=SN1/AlarmList=AL1 HTTP/1.1  Host: example.org  Content-Type: application/json-patch+json  [  {  "op": "add",  "path": "/attributes/alarmRecords/alarmId1/ackUserId",  "value": "userId1"  },  {  "op": "add",  "path": "/attributes/alarmRecords/alarmId1/ackSystemId",  "value": "systemId1"  },  {  "op": "replace",  "path": "/attributes/alarmRecords/alarmId1/ackState",  "value": "ACKNOWLEDGED"  }  ] |

3GPP JSON Patch allows for a more compact request.

|  |
| --- |
| PATCH /SubNetwork=SN1/AlarmList=AL1 HTTP/1.1  Host: example.org  Content-Type: application/vnd.3gpp.json-patch+json  [  {  "op": "merge",  "path": "#/attributes/alarmRecords/alarmId1",  "value": {  "ackUserId": "userId1",  "ackSystemId": "systemId1",  "ackState": "ACKNOWLEDGED"  }  }  ] |

Also JSON Merge Patch is quite compact.

|  |
| --- |
| PATCH /SubNetwork=SN1/AlarmList=AL1 HTTP/1.1  Host: example.org  Content-Type: application/merge-patch+json  {  "id": "AL1",  "attributes": {  "alarmRecords": {  "alarmId1": {  "ackUserId": "userId1",  "ackSystemId": "systemId1",  "ackState": "ACKNOWLEDGED"  }  }  }  } |

**Commenting alarms**

In this example a comment is added to an alarm identified with its "alarmId".

|  |
| --- |
| PATCH /SubNetwork=SN1/AlarmList=AL1 HTTP/1.1  Host: example.org  Content-Type: application/json-patch+json  [  {  "op": "add",  "path": "/attributes/alarmRecords/alarmId1/comments/-",  "value":  {  "commentUserId": "userId1",  "commentSystemId": "systemId1",  "commentText": "Here is the comment text"  }  }  ] |

The MnS producer adds the "commentTime" attribute to the alarm record. The response may be as follows.

|  |
| --- |
| HTTP/1.1 200 OK  Date: Tue, 06 Aug 2019 16:50:26 GMT  Content-Type: application/json  {  "commentTime": "2019-08-06T16:50:26Z",  "commentUserId": "id",  "commentSystemId": "id",  "commentText": "Here is the comment text"  } |

## A.2 RESTful HTTP-based solution set for integration with ONAP VES API

### A.2.1 General

Mapping of Classes, attributes and notifications is identical to those described in clause 10.4.

### A.2.2 Mapping of notifications

#### A.2.2.1 General

The URI of the notification target on the MnS consumer is defined by the notificationRecipientAddress in the NtfSubscriptionControl IOC (See 4.3.22.2 in TS 28.622 [5]. The resource URI is extended with /eventListener.

#### A.2.2.2 Resources

Figure A.2.2.2 -1 shows the resource structure of the fault supervision data report MnS in the context of its integration with VES Event Listener 7.1.1 [9].

**A screen shot of a computer program

Description automatically generated with low confidence**

**Figure A.2.2.2-1: Resource URI structure of the fault supervision data report MnS for integration with ONAP VES Event Listener 7.1.1 (Resource structure section) [9]**

See also Resource structure section in [9].

### A.2.3 Integration with ONAP VES

Detailed guidelines for integration of performance assurance MnS notifications with ONAP VES are provided in Annex B of TS 28.532 [2].

## A.3 NETCONF/YANG solution set

### A.3.1 General

The YANG-Netconf solution set uses the same notifications as OpenAPI, see clause 10.2.

### A.3.2 YANG definitions

YANG definitions for NRM are specified in Forge [11].

Directory: yang-models

Files:

\_3gpp-common-fm.yang

Annex B (informative):   
Probable Causes

This annex lists probable causes.

Sources of these probable causes are ITU-T Recommendation M.3100 [7], ITU-T Recommendation X.721 [6], ITU‑T Recommendation X.733 [8], and ITU-T Recommendation X.736 [13]. In addition, probable causes for wireless systems are listed in ETSI TS 101 251 V6.3.0 (1999-07) [3].

NOTE 1: Probable causes that are defined by more than one standard have been removed to ensure unicity.

Table B.1: Probable Causes from ITU-T Recommendation M.3100 [7]

| **M.3100 Probable cause (string)** | **Event Type** |
| --- | --- |
| Indeterminate | Unknown |
| Alarm Indication Signal (AIS) | Communications |
| Call Setup Failure | Communications |
| Degraded Signal | Communications |
| Far End Receiver Failure (FERF) | Communications |
| Framing Error | Communications |
| Loss Of Frame (LOF) | Communications |
| Loss Of Pointer (LOP) | Communications |
| Loss Of Signal (LOS) | Communications |
| Payload Type Mismatch | Communications |
| NOTE 1: Values 10 correspond to a duplicated probable cause. |  |
| Remote Alarm Interface | Communications |
| Excessive Bit Error Rate (EBER) | Communications |
| Path Trace Mismatch | Communications |
| Unavailable | Communications |
| Signal Label Mismatch | Communications |
| Loss Of Multi Frame | Communications |
| Communications Receive Failure | Communications |
| Communications Transmit Failure | Communications |
| Modulation Failure | Communications |
| Demodulation Failure | Communications |
| NOTE 2: Values 21-26 correspond to duplicated probable causes. |  |
| NOTE 3: Values 27-50 are reserved for M.3100 potential future extensions. |  |
| Back Plane Failure | Equipment |
| Data Set Problem | Equipment |
| Equipment Identifier Duplication | Equipment |
| External IF Device Problem | Equipment |
| Line Card Problem | Equipment |
| Multiplexer Problem | Equipment |
| NE Identifier Duplication | Equipment |
| Power Problem | Equipment |
| Power Supply Failure | Equipment |
| Processor Problem | Equipment |
| Protection Path Failure | Equipment |
| Receiver Failure | Equipment |
| Replaceable Unit Missing | Equipment |
| Replaceable Unit Type Mismatch | Equipment |
| Synchronization Source Mismatch | Equipment |
| Terminal Problem | Equipment |
| Timing Problem | Equipment |
| Transmitter Failure | Equipment |
| Trunk Card Problem | Equipment |
| Replaceable Unit Problem | Equipment |
| Real Time Clock Failure | Equipment |
| NOTE 4: Values 71-80 correspond to duplicated values. |  |
| Protection Mechanism Failure | Equipment |
| Protecting Resource Failure | Equipment |
| NOTE 5: Values 83-100 are reserved for M.3100 potential future extensions. |  |
| Air Compressor Failure | Environmental |
| Air Conditioning Failure | Environmental |
| Air Dryer Failure | Environmental |
| Battery Discharging | Environmental |
| Battery Failure | Environmental |
| Commercial Power Failure | Environmental |
| Cooling Fan Failure | Environmental |
| Engine Failure | Environmental |
| Fire Detector Failure | Environmental |
| Fuse Failure | Environmental |
| Generator Failure | Environmental |
| Low Battery Threshold | Environmental |
| Pump Failure | Environmental |
| Rectifier Failure | Environmental |
| Rectifier High Voltage | Environmental |
| Rectifier Low F Voltage | Environmental |
| Ventilation System Failure | Environmental |
| Enclosure Door Open | Environmental |
| Explosive Gas | Environmental |
| Fire | Environmental |
| Flood | Environmental |
| High Humidity | Environmental |
| High Temperature | Environmental |
| High Wind | Environmental |
| Ice Build Up | Environmental |
| Intrusion Detection | Environmental |
| Low Fuel | Environmental |
| Low Humidity | Environmental |
| Low Cable Pressure | Environmental |
| Low Temperature | Environmental |
| Low Water | Environmental |
| Smoke | Environmental |
| Toxic Gas | Environmental |
| NOTE 6: Values 137-150 are reserved for M.3100 potential future extensions. |  |
| Storage Capacity Problem | Processing Error |
| Memory Mismatch | Processing Error |
| Corrupt Data | Processing Error |
| Out Of CPU Cycles | Processing Error |
| Software Environment Problem | Processing Error |
| Software Download Failure | Processing Error |
| Loss of Real Time | Processing Error |
| Reinitialized | Processing Error |
| NOTE 7: Values 159-167 correspond to duplicated probable causes. |  |
| NOTE 8: Values 168-200 are reserved for M.3100 potential future extensions. |  |
| NOTE 9: Values 201-202 correspond to duplicated probable causes. |  |
| Excessive Error Rate | Quality of service |
| NOTE 10: Values 204-207 correspond to duplicated probable causes. |  |
| NOTE 11: Values 208-300 are reserved for M.3100 potential future extensions. |  |

Table B.2: Probable Causes from ITU-T Recommendation X.721 [6], X.733 [8], X.736 [13]

| **X.721/X.733/X.736 Probable Cause (string)** | **Even Type** |
| --- | --- |
| Adapter Error | Equipment |
| Application Subsystem Failure | Processing error |
| Bandwidth Reduction | Security Service or Mechanism Violation |
| NOTE 1: Values 304 correspond to duplicated probable cause. |  |
| Communication Protocol Error | Communications |
| Communication Subsystem Failure | Communications |
| Configuration or Customizing Error | Processing error |
| Congestion | Quality of service |
| NOTE 2: Values 309 correspond to duplicated probable cause. |  |
| CPU Cycles Limit Exceeded | Processing error |
| Data Set or Modem Error | Equipment |
| NOTE 3: Values 312 correspond to duplicated probable cause. |  |
| DTE-DCE Interface Error | Communications |
| NOTE 4: Values 314 correspond to duplicated probable cause. |  |
| Equipment Malfunction | Communications |
| Excessive Vibration | Integrity Violation |
| File Error | Environmental |
| NOTE 5: Values 318-320 correspond to duplicated probable cause. | Equipment |
| Heating or Ventilation or Cooling System Problem | Environmental |
| Humidity Unacceptable | Environmental |
| Input/Output Device Error | Equipment |
| Input Device Error | Environmental |
| LAN Error | Processing error |
| Leak Detection | Environmental |
| Local Node Transmission Error | Communications |
| NOTE 6: Values 328-329 correspond to duplicated probable cause. |  |
| Material Supply Exhausted | Environmental |
| NOTE 7: Values 331 correspond to duplicated probable cause. |  |
| Out of Memory | Processing error |
| Output Device Error | Equipment |
| Performance Degraded | Quality of service |
| NOTE 8: Values 335 correspond to duplicated probable cause. |  |
| Pressure Unacceptable | Operational Violation |
| NOTE 9: Values 337-338 correspond to duplicated probable cause. |  |
| Queue Size Exceeded | Quality of service |
| Receive Failure | Equipment |
| NOTE 10: Values 341 correspond to duplicated probable cause. |  |
| Remote Node Transmission Error | Communications |
| Resource at or Nearing Capacity | Quality of service |
| Response Time Excessive | Quality of service |
| Re-transmission Rate Excessive | Quality of service |
| Software Error | Processing error |
| Software Program Abnormally Terminated | Processing error |
| Software Program Error | Processing error |
| NOTE 11: Values 349 correspond to duplicated probable cause. |  |
| Temperature Unacceptable | Environmental |
| Threshold Crossed | Quality of service |
| NOTE 12: Values 352 correspond to duplicated probable cause. |  |
| Toxic Leak Detected | Environmental |
| Transmit Failure | Equipment |
| NOTE 13: Values 355 correspond to duplicated probable cause. |  |
| Underlying Resource Unavailable | Processing error |
| Version Mismatch | Processing error |
| NOTE 14: Values 358-500 are reserved for X.721 potential future extensions. |  |

Table B.3: Probable Causes for Wireless Systems from ETSI TS 101 251 V6.3.0 (1999-07) [3]

| **Wireless Systems (string)** | **Event Type** |
| --- | --- |
| A-bis to BTS interface failure | Equipment |
| A-bis to TRX interface failure | Equipment |
| Antenna problem | Equipment |
| Battery breakdown | Equipment |
| Battery charging fault | Equipment |
| Clock synchronization problem | Equipment |
| Combiner problem | Equipment |
| Disk problem | Equipment |
| NOTE 1 Values 509 correspond to duplicated probable cause. |  |
| Excessive receiver temperature | Equipment |
| Excessive transmitter output power | Equipment |
| Excessive transmitter temperature | Equipment |
| Frequency hopping degraded | Equipment |
| Frequency hopping failure | Equipment |
| Frequency redefinition failed | Equipment |
| Line interface failure | Equipment |
| Link failure | Equipment |
| Loss of synchronization | Equipment |
| Lost redundancy | Equipment |
| Mains breakdown with battery back-up | Equipment |
| Mains breakdown without battery back-up | Equipment |
| Power supply failure | Equipment |
| Receiver antenna fault | Equipment |
| NOTE 2: Values 524 correspond to duplicated probable cause. |  |
| Receiver multicoupler failure | Equipment |
| Reduced transmitter output power | Equipment |
| Signal quality evaluation fault | Equipment |
| Timeslot hardware failure | Equipment |
| Transceiver problem | Equipment |
| Transcoder problem | Equipment |
| Transcoder or rate adapter problem | Equipment |
| Transmitter antenna failure | Equipment |
| Transmitter antenna not adjusted | Equipment |
| NOTE 3: Values 534 correspond to duplicated probable cause. |  |
| Transmitter low voltage or current | Equipment |
| Transmitter off frequency | Equipment |
| Database inconsistency | Processing error |
| File system call unsuccessful | Processing error |
| Input parameter out of range | Processing error |
| Invalid parameter | Processing error |
| Invalid pointer | Processing error |
| Message not expected | Processing error |
| Message not initialized | Processing error |
| Message out of sequence | Processing error |
| System call unsuccessful | Processing error |
| Timeout expired | Processing error |
| Variable out of range | Processing error |
| Watch dog timer expired | Processing error |
| Cooling system failure | Environmental |
| External equipment failure | Environmental |
| External power supply failure | Environmental |
| External transmission device failure | Environmental |
| NOTE 4: Values 553-560 correspond to duplicated probable cause. |  |
| Reduced alarm reporting | Quality of service |
| Reduced event reporting | Quality of service |
| Reduced logging capability | Quality of service |
| System resources overload | Quality of service |
| Broadcast channel failure | Communications |
| Connection establishment error | Communications |
| Invalid message received | Communications |
| Invalid MSU received | Communications |
| LAPD link protocol failure | Communications |
| Local alarm indication | Communications |
| Remote alarm indication | Communications |
| Routing failure | Communications |
| SS7 protocol failure | Communications |
| Transmission error | Communications |
| NOTE 5: Values 575 correspond to duplicated probable cause. |  |
| NOTE 6: Values 576-700 are reserved for ETSI potential future extensions. |  |

Table B.4: Probable Causes for Security Alarm from M3100 X.736 [13]

| **Wireless Systems (string)** | **Even Type** |
| --- | --- |
| Authentication Failure | security service or mechanism violation |
| Breach of Confidentiality | security service or mechanism violation |
| Cable Tamper | physical violation |
| Delayed Information | time domain violation |
| Denial of Service | operational violation |
| Duplicate Information | integrity violation |
| Information Missing | integrity violation |
| Information Modification Detected | integrity violation |
| Information Out of Sequence | integrity violation |
| Intrusion Detection | physical violation |
| Key Expired | time domain violation |
| Non Repudiation Failure | security service or mechanism violation |
| Out of Hours Activity | time domain violation |
| Out of Service | operational violation |
| Procedural Error | operational violation |
| Unauthorised Access Attempt | security service or mechanism violation |
| Unexpected Information | integrity violation |
| Unspecified Reason | security service or mechanism violation |
| NOTE: Values 719-800 are reserved for M.3100 potential future extensions. | |

Annex C (informative):  
Change history

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Change history** | | | | | | | |
| **Date** | **Meeting** | **TDoc** | **CR** | **Rev** | **Cat** | **Subject/Comment** | **New version** |
| 2023--08 | SA5#150 | S5-235867 | - | - | - | Initial skeleton | 0.0.0 |
| 2023--08 | SA5#150 | S5-235093 | - | - | - | Rel-18 pCR 28.111 FM service first draft | 0.1.0 |
| 2023--08 | SA5#150 | S5-235106 | - | - | - | Rel-18 pCR 28.111 FM service, definition updates | 0.1.0 |
| 2023-10 | SA5#151 | S5-237046 | - | - | - | Remove requirements on virtual resources  Stage 3 Code to be specified in Forge. | 0.2.0 |
| 2023-11 | SA5#152 | S5-238142 |  |  |  | Rel-18 pCR 28.111 FM full updates | 0.3.0 |
| 2023-12 | SA#102 | SP-231526 |  |  |  | Presented for information | 1.0.0 |
| 2024-01 | SA5#153 | S5-240973 |  |  |  | Rel-18 pCR 28.111 FM full updates | 1.1.0 |
| 2024-03 | SA#103 | SP-240259 |  |  |  | Presented for approval | 2.0.0 |
| 2024-03 | SA#103 |  |  |  |  | Upgrade to change control version | 18.0.0 |