**3GPP TSG- Meeting # *939***

**, , - revision of S5-253277**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| *CR-Form-v12.3* | | | | | | | | |
| **CHANGE REQUEST** | | | | | | | | |
|  | | | | | | | | |
|  |  | **CR** |  | **rev** | **1** | **Current version:** |  |  |
|  | | | | | | | | |
| *For* [***HE******LP***](http://www.3gpp.org/3G_Specs/CRs.htm#_blank)*on using this form: comprehensive instructions can be found at* [*http://www.3gpp.org/Change-Requests*](http://www.3gpp.org/Change-Requests)*.* | | | | | | | | |
|  | | | | | | | | |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ***Proposed change affects:*** | UICC apps |  | ME |  | Radio Access Network | **X** | Core Network | **X** |

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | | | | | |
| ***Title:*** |  | | | | | | | | | |
|  |  | | | | | | | | | |
| ***Source to WG:*** |  | | | | | | | | | |
| ***Source to TSG:*** | S5 | | | | | | | | | |
|  |  | | | | | | | | | |
| ***Work item code:*** | \_Ph3 | | | | |  | ***Date:*** | | |  |
|  |  | | | |  | |  | | |  |
| ***Category:*** | F |  | | | | | ***Release:*** | | |  |
|  | *Use one of the following categories:* ***F*** *(correction)* ***A*** *(mirror corresponding to a change in an earlier release)* ***B*** *(addition of feature),* ***C*** *(functional modification of feature)* ***D*** *(editorial modification)*  Detailed explanations of the above categories can be found in 3GPP [TR 21.900](http://www.3gpp.org/ftp/Specs/html-info/21900.htm). | | | | | | | | *Use one of the following releases: Rel-8 (Release 8) Rel-9 (Release 9) Rel-10 (Release 10) Rel-11 (Release 11) … Rel-17 (Release 17) Rel-18 (Release 18) Rel-19 (Release 19)  Rel-20 (Release 20)* | |
|  |  | | | | | | | | | |
| ***Reason for change:*** | | FM handles both ADAC (Automatically Detected and Automatically Cleared) and ADMC (Automatically Detected and Manually Cleared) alarms. It is not visible for the consumer whether an individual alarm needs manual clearing or not. If an ADMC is not cleared it can stay in the alarm list for an excessive length of time. To avoid this the consumer should get an indication if an alarm is ADMC.  The case when the alarmlist becomes full shall be described.  As the size of the alarm list is finite, if it becomes full, the producer may remove the oldest alarms. If there are cleared but unacknowledged alarms, these should be removed before any not-cleared alarms are removed. | | | | | | | | |
|  | |  | | | | | | | | |
| ***Summary of change:*** | | Add ADMC indication to the alarmrecord.  Clarify behavior in case of full alarm list.  Typo corrections | | | | | | | | |
|  | |  | | | | | | | | |
| ***Consequences if not approved:*** | | If the user is not aware that an alarm is ADMC, it might never get cleared.  Unclear behavior when alarm list is full. | | | | | | | | |
|  | |  | | | | | | | | |
| ***Clauses affected:*** | | 6.5, 6.8, 7.3.1.2, 7.3.1.3, 7.3.2, 7.3.2.1, 7.4.1, 8.8.2, A.1.4, Forge | | | | | | | | |
|  | |  | | | | | | | | |
|  | | **Y** | **N** |  | | | |  | | |
| ***Other specs*** | |  | **X** | Other core specifications | | | | TS/TR ... CR ... | | |
| ***affected:*** | |  | **X** | Test specifications | | | | TS/TR ... CR ... | | |
| ***(show related CRs)*** | |  | **X** | O&M Specifications | | | | TS/TR ... CR ... | | |
|  | |  | | | | | | | | |
| ***Other comments:*** | | YANG Forge MR link: <https://forge.3gpp.org/rep/sa5/MnS/-/merge_requests/1795> at commit 28580d99c9097d9c2d3fea7824a4aafa8e24e60f  YAMl: Forge MR link: <https://forge.3gpp.org/rep/sa5/MnS/-/merge_requests/1821> at commit 64ff87a63e0b73a144f903d776b98eebd35cf800 | | | | | | | | |
|  | |  | | | | | | | | |
| ***This CR's revision history:*** | |  | | | | | | | | |

***First change***

## 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 managed 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.

***Next change***

## 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 (for ADMC alarms) 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.

ADAC alarms cannot be cleared by the consumer.

***Next change***

#### 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 |
| clearingType | CO | T | F | T | 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 shall 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. | |
| clearingType | This attribute should be supported in case ADMC alarms may be raised by the system. | |

***Next change***

### 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 can neither 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.

As the size of the alarm list is finite, if it becomes full, the producer may remove the oldest list entries in alarmRecords. If there are cleared but unacknowledged alarms these shall be removed before any not-cleared alarms are removed.

***Next change***

### 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** |
| --- | --- | --- |
| objectInstance | Managed object instance identified by its DN. | type: DN  multiplicity: 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 managed 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  defaultValue: 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(s) of the alarm scope that may not be reliable.  Case of the complete alarm list is unreliable:  If this parameter is equal to the instance carried in systemDN, then all AlarmRecord instances in the AlarmList may not be reliable.    Case of one or more unreliable subtrees of the complete alarm list is (are) unreliable:  If this parameter is equal to some instance(s) represented by MonitoredEntity, then only AlarmRecord related to this instance(s) 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 AlarmRecord is changed by the alarmed resource. Changes to AlarmRecord 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.  Other:  The type of the alarm does not fit into any of the above types or is not known.  allowedValues:  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  OTHER | 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 B for a complete listing. The producer should choose the most specific probableCause applicable. | 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: ThresholdCrossing  multiplicity: \*  isOrdered: False  isUnique: True defaultValue: None  isNullable: False |
| observedMeasurement | The name of the monitored measurement that crossed the threshold and that caused the notification (Rec. ITU-T X. 733 [8]). | type: string  multiplicity: 1  isOrdered: N/A  isUnique: N/A defaultValue: None  isNullable: False |
| observedValue | The value of the gauge or counter which crossed the threshold. This may be different from the threshold value if, for example, the gauge may only take on discrete values. Integer values are used for counters and float values for gauges (Rec. ITU-T X. 733 [8]). Note that a "number" type property can contain both integers and floating point numbers. | type: number  multiplicity: 1  isOrdered: N/A  isUnique: N/A defaultValue: None  isNullable: False |
| thresholdLevel | In the case of a gauge the threshold level specifies a pair of threshold values, the first being the value of the crossed threshold and the second, its corresponding hysteresis; in the case of a counter the threshold level specifies only the threshold value (Rec. ITU-T X. 733 [8]). | type: ThresholdLevelInd  multiplicity: 0..1  isOrdered: N/A  isUnique: N/A defaultValue: None  isNullable: False |
| up | Indicates for counter and gauge thresholds that the threshold crossing occurred when going up. | type: ThresholdHysteresis  multiplicity: 0..1  isOrdered: N/A  isUnique: True defaultValue: None  isNullable: False |
| down | Indicates for gauge thresholds that the threshold crossing occurred when going down, applicable only to gauge thresholds. | type: ThresholdHysteresis  multiplicity: 0..1  isOrdered: N/A  isUnique: N/A defaultValue: None  isNullable: False |
| high | Higher value of a threshold with hysteresis, the integer type is used for counter thresholds and the float type for gauge thresholds. | type: integer or Float  multiplicity: 1  isOrdered: N/A  isUnique: N/A defaultValue: None  isNullable: False |
| low | Lower value of a threshold with hysteresis, applicable only to gauge thresholds. | type: Float  multiplicity: 0..1  isOrdered: N/A  isUnique: N/A defaultValue: None  isNullable: False |
| armTime | For a gauge threshold, the time at which the threshold was last re-armed, namely the time after the previous threshold crossing at which the hysteresis value of the threshold was exceeded thus again permitting generation of notifications when the threshold is crossed. For a counter threshold, the later of the time at which the threshold offset was last applied, or the time at which the counter was last initialized (for resettable counters) (Rec. ITU-T X. 733 [8]) | type: DateTime  multiplicity: 0..1  isOrdered: N/A  isUnique: N/A 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 record. 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 are 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 AlarmRecord 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 |
| clearingType | Indicates whether the alarm needs to be cleared manually by the MnS consumer (ADMC) or the producer will clear it automatically (ADAC).  AllowedValues:  MANUAL, AUTOMATIC | type: ENUM  multiplicity: 1  isOrdered: N/A  isUnique: N/A  defaultValue: AUTOMATIC  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.notificationIds | A list of correlated notificationIds. | type: integer  multiplicity: 1..\*  isOrdered: False  isUnique: True defaultValue: None  isNullable: False |
| NOTEs: none. | | |

***Next change***

### 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 | See TS 28.532 [2] clause 11.0.2 |  |
| objectInstance | M | alarmRecord.objectInstance  DN of the MonitoredEntity that is the source of the alarm |  |
| notificationId | M | See TS 28.532 [2] clause 11.0.2 |  |
| notificationType | M | "notifyNewAlarm" |  |
| eventTime | M | alarmRecord.alarmRaisedTime |  |
| systemDN | M | See TS 28.532 [2] clause 11.0.2 |  |
| sequenceNo | CM | See TS 28.532 [2] clause 11.0.2 |  |
| subscriptionId | CM | See TS 28.532 [2] clause 11.0.2 |  |
| 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. |
| clearingType | CO | alarmRecord.clearingType | This parameter should be supported in case ADMC alarms may be raised by the system. |

***Next change***

### A.1.4 Examples

**Sending alarm notifications**

This example shows how a "notifyNewAlarm" notification is sent.

|  |
| --- |
| POST /3gpp-management/alarm-notification-sink HTTP/1.1  Host: example.org  Content-Type: application/json  {  "href": "https://example.org/SubNetwork=SN1/ManagedElement=ME1",  "notificationId": 123456789,  "notificationType": "notifyNewAlarm",  "eventTime": "2024-08-21T16:39:57-08:00",  "systemDN": "DC=example.org,SubNetwork=SN1,MnsAgent=MA1",  "alarmId": "alarm-id-1",  "alarmType": "EQUIPMENT\_ALARM",  "probableCause": "Indeterminate",  "perceivedSeverity": "CRITICAL"  } |

**Retrieving alarms**

This 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?\  filter=/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?\  filter=/AlarmList[id="AL1"]/attributes/alarmRecords\  /\*[objectInstance="DN1"] HTTP/1.1 |

A MnS consumer wants to retrieve often all alarms from one Managed Element. A Managed 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?\  filter=/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"  } |

***Next change***

Forge MR link: <https://forge.3gpp.org/rep/sa5/MnS/-/merge_requests/1795> at commit 28580d99c9097d9c2d3fea7824a4aafa8e24e60f

\*\*\* START OF CHANGE 1 \*\*\*

\*\*\* yang-models/\_3gpp-common-fm.yang \*\*\*

<CODE BEGINS>

module \_3gpp-common-fm {

yang-version 1.1;

namespace "urn:3gpp:sa5:\_3gpp-common-fm";

prefix "fm3gpp";

import ietf-yang-types { prefix yang; }

import \_3gpp-common-top { prefix top3gpp; }

import \_3gpp-common-yang-types { prefix types3gpp; }

import \_3gpp-common-yang-extensions { prefix yext3gpp; }

organization "3GPP SA5";

contact "https://www.3gpp.org/DynaReport/TSG-WG--S5--officials.htm?Itemid=464";

description "Defines a Fault Management model

Copyright 2025, 3GPP Organizational Partners (ARIB, ATIS, CCSA, ETSI, TSDSI,

TTA, TTC). All rights reserved.";

reference "3GPP TS 28.111";

revision 2025-07-01 { reference "CR-0050 CR-0051"; }

revision 2025-05-01 { reference "CR-0042 CR-0043"; } // common for R18, R19

revision 2025-03-25 { reference "CR-0025 CR-0026"; }

revision 2024-05-12 {

description "The definition of the module was from TS 28.623 to TS 28.111";

reference CR-0008 ;

}

revision 2024-03-06 { reference CR-0333 ; }

revision 2024-02-24 { reference CR-0346; }

revision 2024-01-18 {

description "The specification of the file is moved from 28.623 to 28.532";

reference "28.623 CR-0315";

}

revision 2023-09-18 { reference CR-0271; }

revision 2023-05-10 { reference CR-0250; }

revision 2022-10-24 { reference CR-0196; }

revision 2021-08-08 { reference "CR-0132"; }

revision 2021-06-02 { reference "CR-0130"; }

revision 2020-06-03 { reference "CR-0091"; }

revision 2020-02-24 { reference "S5-201365"; }

feature AcknowledgeByConsumer {

description "Indicates whether alarm acknowledgement by the consumer is

supported.";

}

typedef eventType {

type enumeration {

enum OTHER {

value 1;

}

enum COMMUNICATIONS\_ALARM {

value 2;

}

enum QUALITY\_OF\_SERVICE\_ALARM {

value 3;

}

enum PROCESSING\_ERROR\_ALARM {

value 4;

}

enum EQUIPMENT\_ALARM {

value 5;

}

enum ENVIRONMENTAL\_ALARM {

value 6;

}

enum INTEGRITY\_VIOLATION {

value 7;

}

enum OPERATIONAL\_VIOLATION {

value 8;

}

enum PHYSICAL\_VIOLATION {

value 9;

}

enum SECURITY\_SERVICE\_OR\_MECHANISM\_VIOLATION {

value 10;

}

enum TIME\_DOMAIN\_VIOLATION {

value 11;

}

}

description "General category for the alarm.";

}

typedef severity-level {

type enumeration {

enum CRITICAL { value 3; }

enum MAJOR { value 4; }

enum MINOR { value 5; }

enum WARNING { value 6; }

enum INDETERMINATE { value 7; }

enum CLEARED { value 8; }

}

description "The possible alarm severities";

}

grouping AlarmCommentGrp {

leaf commentTime {

type yang:date-and-time;

config false;

mandatory true;

yext3gpp:inVariant;

description "Date and Time the comment was created.";

}

leaf commentUserId {

type string;

mandatory true;

yext3gpp:inVariant;

description "It carries the identification of the user who made the

comment.";

}

leaf commentSystemId {

type string;

mandatory true;

yext3gpp:inVariant;

description "It carries the identification of the system (

Management System) from which the comment is made. That system

supports the user that made the comment.";

}

leaf commentText {

type string;

mandatory true;

yext3gpp:inVariant;

description "It carries the textual comment.";

}

}

grouping ThresholdHysteresisGrp {

description "The ThresholdHysteresis defines the threshold boundaries to

control the hysteresis mechanism.

The high attribute of ThresholdHysteresis identifies the higher value of

a threshold with hysteris, the integer type is used for counter

thresholds and the float type for gauge thresholds. The low attribute

of ThresholdHysteresis identifies the lower value of a threshold with

hysteresis, applicable only to gauge thresholds.";

leaf high {

type union {

type int64;

type decimal64 {

fraction-digits 7;

}

}

mandatory true;

description "Higher value of a threshold with hysteresis, the integer

type is used for counter thresholds and the float type for gauge

thresholds.";

}

leaf low {

type decimal64 {

fraction-digits 7;

}

description "Lower value of a threshold with hysteresis, applicable

only to gauge thresholds.";

}

}

grouping ThresholdLevelIndGrp {

description "The up attribute indicates for counter and gauge thresholds

that the threshold crossing occurred when going up. The down attribute

only indicates for gauge thresholds that the threshold crossing occurred

when going down, applicable only to gauge thresholds.";

list up {

description "Indicates for counter and gauge thresholds that the

threshold crossing occurred when going up.";

max-elements 1;

key idx;

leaf idx { type int32; }

uses ThresholdHysteresisGrp;

}

list down {

description "Indicates for gauge thresholds that the threshold crossing

occurred when going down, applicable only to gauge thresholds.";

max-elements 1;

key idx;

leaf idx { type int32; }

uses ThresholdHysteresisGrp;

}

}

grouping ThresholdCrossingGrp {

description "The datatype indicates the crossed threshold

information regardless of the gauge threshold, which represents an

instantaneous value that changes over time, or the counter threshold,

which represents monotonically increasing cumulative quantity.

The observedMeasurement attribute of TheresholdInfo specifies the name

of the monitored measurement that crossed the threshold and that

caused the notification (Rec. ITU-T X. 733[8]). The observedValue

attribute indicates the value of the gauge or counter which crossed

the threshold. This may be different from the threshold value if, for

example, the gauge may only take on discrete values. Integer values

are used for counters and float values for gauges (Rec. ITU-T X. 733).

Note that a 'number' type property can contain both integers and

floating point numbers.

For the thresholdLevel attribute, in the case of a gauge, it specifies

a pair of threshold values, the first being the value of the crossed

threshold and the second, its corresponding hysteresis; in the case of

a counter, it specifies only the threshold value (Rec. ITU-T X. 733).

For the armTime attribute, for a gauge threshold, it specifies the

time at which the threshold was last re-armed, namely the time after

the previous threshold crossing at which the hysteresis value of the

threshold was exceeded thus again permitting generation of

notifications when the threshold is crossed; for a counter threshold,

the later of the time at which the threshold offset was last applied,

or the time at which the counter was last initialized (for resettable

counters) (Rec. ITU-T X. 733).";

leaf observedMeasurement {

type string;

mandatory true;

description "The name of the monitored measurement that crossed the

threshold and that caused the notification (Rec. ITU-T X. 733 ";

}

leaf observedValue {

type union {

type int64;

type decimal64 {

fraction-digits 7;

}

}

mandatory true;

description "The value of the gauge or counter which crossed the

threshold. This may be different from the threshold value if, for

example, the gauge may only take on discrete values.

Integer values are used for counters and float values for gauges

(Rec. ITU-T X. 733). Note that a 'number' type property can contain

both integers and floating point numbers.";

}

list thresholdLevel {

description "In the case of a gauge the threshold level specifies

a pair of threshold values, the first being the value of the crossed

threshold and the second, its corresponding hysteresis; in the case

of a counter the threshold level specifies only the threshold value

(Rec. ITU-T X. 733).";

max-elements 1;

key idx;

leaf idx { type int32; }

uses ThresholdLevelIndGrp;

}

leaf armTime {

type yang:date-and-time;

description "For a gauge threshold, the time at which the threshold

was last re-armed, namely the time after the previous threshold

crossing at which the hysteresis value of the threshold was

exceeded thus again permitting generation of notifications when the

threshold is crossed. For a counter threshold, the later of the time

at which the threshold offset was last applied, or the time at

which the counter was last initialized (for resettable counters)

(Rec. ITU-T X. 733)";

}

}

grouping AlarmRecordGrp {

description "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 ACKNOWLEDED.";

leaf alarmId {

type string;

mandatory true;

description "Identifies the alarmRecord";

yext3gpp:notNotifyable;

yext3gpp:inVariant;

}

leaf objectInstance {

type types3gpp:DistinguishedName;

config false ;

mandatory true;

yext3gpp:notNotifyable;

yext3gpp:inVariant;

}

leaf notificationId {

type int32;

config false ;

mandatory true;

description "The Id of the last notification updating the AlarmRecord.";

yext3gpp:notNotifyable;

}

leaf alarmRaisedTime {

type yang:date-and-time ;

mandatory true;

config false ;

yext3gpp:notNotifyable;

}

leaf alarmChangedTime {

type yang:date-and-time ;

config false ;

description "not applicable if related alarm has not changed";

yext3gpp:notNotifyable;

}

leaf alarmClearedTime {

type yang:date-and-time ;

config false ;

description "not applicable if related alarm was not cleared";

yext3gpp:notNotifyable;

}

leaf alarmType {

type eventType;

config false ;

mandatory true;

description "General category for the alarm.";

yext3gpp:notNotifyable;

yext3gpp:inVariant;

}

leaf probableCause {

type union {

type int32;

type string;

}

config false ;

mandatory true;

yext3gpp:notNotifyable;

yext3gpp:inVariant;

}

leaf specificProblem {

type union {

type int32;

type string;

}

config false ;

reference "ITU-T Recommendation X.733 clause 8.1.2.2.";

yext3gpp:notNotifyable;

yext3gpp:inVariant;

}

leaf perceivedSeverity {

type severity-level;

mandatory true;

description "This is Writable only if producer supports consumer

to set perceivedSeverity to CLEARED";

yext3gpp:notNotifyable;

}

leaf backedUpStatus {

type boolean;

config false ;

description "Indicates if an object (the MonitoredEntity) has a back

up. See definition in ITU-T Recommendation X.733 clause 8.1.2.4.";

yext3gpp:notNotifyable;

}

leaf backUpObject {

type types3gpp:DistinguishedName;

config false ;

description "Backup object of the alarmed object as defined in

ITU-T Rec. X. 733";

yext3gpp:notNotifyable;

}

leaf trendIndication {

type enumeration {

enum MORE\_SEVERE;

enum NO\_CHANGE;

enum LESS\_SEVERE;

}

config false ;

description "Indicates if some observed condition is getting better,

worse, or not changing. ";

reference "ITU-T Recommendation X.733 clause 8.1.2.6.";

yext3gpp:notNotifyable;

}

list thresholdInfo {

config false ;

yext3gpp:notNotifyable;

description "Indicates the crossed threshold";

key idx;

leaf idx { type int32; }

uses ThresholdCrossingGrp;

}

list stateChangeDefinition {

key attributeName;

config false ;

description "Indicates MO 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 [4] clause 8.1.2.10.";

yext3gpp:notNotifyable;

leaf attributeName {

type string;

}

anydata newValue {

mandatory true;

description "The new value of the attribute. The content of this data

node shall be in accordance with the data model for the attribute.";

}

anydata oldValue{

description "The old value of the attribute. The content of this data

node shall be in accordance with the data model for the attribute.";

}

}

list monitoredAttributes {

key attributeName;

config false ;

yext3gpp:notNotifyable;

description "Attributes of the monitored entity and their

values at the time the alarm occurred that are of interest for the

alarm report.";

reference "ITU-T Recommendation X.733 clause 8.1.2.11.";

leaf attributeName {

type string;

}

anydata value {

mandatory true;

description "The value of the attribute. The content of this data

node shall be in accordance with the data model for the attribute.";

}

}

leaf proposedRepairActions {

type string;

config false ;

description "Indicates proposed repair actions. See definition in

ITU-T Recommendation X.733 clause 8.1.2.12.";

yext3gpp:notNotifyable;

}

leaf additionalText {

type string;

config false ;

yext3gpp:notNotifyable;

}

list additionalInformation {

key name;

config false ;

yext3gpp:notNotifyable;

description "Vendor specific alarm information in the alarm.";

uses types3gpp:nameValuePair;

}

leaf rootCauseIndicator {

type boolean;

default false;

config false ;

description "It indicates that this AlarmInformation is the root cause

of the events captured by the notifications whose identifiers are in

the related CorrelatedNotification instances.";

yext3gpp:notNotifyable;

}

list comments {

yext3gpp:notNotifyable;

description "List of comments and data about the comments.";

key idx;

leaf idx { type uint32; }

uses AlarmCommentGrp;

}

leaf ackTime {

if-feature AcknowledgeByConsumer;

type yang:date-and-time ;

config false ;

description "It identifies the time when the alarm has been

acknowledged or unacknowledged the last time, i.e. it registers the

time when ackState changes.";

yext3gpp:notNotifyable;

}

leaf ackUserId {

if-feature AcknowledgeByConsumer;

type string;

description "It identifies the last user who has changed the

Acknowledgement State.";

yext3gpp:notNotifyable;

}

leaf ackSystemId {

if-feature AcknowledgeByConsumer;

type string;

description "It identifies the system (Management System) that last

changed the ackState of an alarm, i.e. acknowledged or unacknowledged

the alarm.";

yext3gpp:notNotifyable;

}

leaf ackState {

if-feature AcknowledgeByConsumer;

type enumeration {

enum ACKNOWLEDGED {

description "The alarm has been acknowledged.";

}

enum UNACKNOWLEDGED {

description "The alarm has unacknowledged or the alarm has never

been acknowledged.";

}

}

yext3gpp:notNotifyable;

}

leaf clearUserId {

type string;

description "Carries the identity of the user who invokes the

clearAlarms operation.";

yext3gpp:notNotifyable;

}

leaf clearSystemId {

type string;

yext3gpp:notNotifyable;

}

leaf serviceUser {

type string;

config false ;

description "It identifies the service-user whose request for service

provided by the serviceProvider led to the generation of the

security alarm.";

yext3gpp:notNotifyable;

}

leaf serviceProvider {

type string;

config false ;

description "It identifies the service-provider whose service is

requested by the serviceUser and the service request provokes the

generation of the security alarm.";

yext3gpp:notNotifyable;

}

leaf securityAlarmDetector {

type string;

config false ;

yext3gpp:notNotifyable;

}

list correlatedNotifications {

key sourceObjectInstance;

description "List of correlated notifications";

leaf sourceObjectInstance {

type types3gpp:DistinguishedName;

}

leaf-list notificationIds {

type int32;

min-elements 1;

}

}

leaf clearingType {

type enumeration {

enum AUTOMATIC;

enum MANUAL;

}

config false;

default AUTOMATIC;

description "Indicates whether the alarm needs

to be cleared manually by the MnS consumer (ADMC)

or the producer will clear it automatically (ADAC).";

}

}

grouping AlarmListGrp {

description "Represents the AlarmList IOC.";

leaf administrativeState {

type types3gpp:BasicAdministrativeState ;

default LOCKED;

description "When set to UNLOCKED, the alarm list is updated.

When the set to LOCKED, the existing alarm records are not

updated, and new alarm records are not added to the alarm list.";

}

leaf operationalState {

type types3gpp:OperationalState ;

default DISABLED;

config false;

description "The producer sets this attribute to ENABLED, indicating

that it has the resource and ability to record alarm in AlarmList

else, it sets the attribute to DISABLED.";

}

leaf numOfAlarmRecords {

type uint32 ;

config false;

mandatory true;

description "The number of alarm records in the AlarmList";

yext3gpp:notNotifyable;

}

leaf lastModification {

type yang:date-and-time ;

config false;

description "The last time when an alarm record was modified";

yext3gpp:notNotifyable;

}

list alarmRecords {

key alarmId;

description "List of alarmRecords";

yext3gpp:notNotifyable;

uses AlarmRecordGrp;

}

leaf-list unreliableAlarmScope {

type types3gpp:DistinguishedName;

config false;

yext3gpp:notNotifyable;

description "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.";

}

}

grouping FmSubtree {

description "Contains FM related classes.

Should be used in all classes (or classes inheriting from)

- SubNetwork

- ManagedElement

If some YAM wants to augment these classes/list/groupings they must

augment all user classes!";

list AlarmList {

key id;

max-elements 1;

yext3gpp:only-system-created;

description "The AlarmList represents the capability to store and manage

alarm records. 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 instances are created by the system or are pre-installed.

They can neighter be created nor deleted by MnS consumers.

When the alarm list is locked or disabled, the existing alarm records

are not updated, and new alarm records are not added to the alarm list

As the size of the alarm list is finite, if it becomes full, the

producer may remove the oldest list entries in alarmRecords. If there

are cleared but unacknowledged alarms these shall be removed before any

not-cleared alarms are removed.";

uses top3gpp:Top\_Grp ;

container attributes {

uses AlarmListGrp ;

}

}

}

}

<CODE ENDS>

\*\*\* END OF CHANGE 1 \*\*\*

Forge MR link: <https://forge.3gpp.org/rep/sa5/MnS/-/merge_requests/1821> at commit 64ff87a63e0b73a144f903d776b98eebd35cf800

\*\*\* START OF CHANGE 1 \*\*\*

\*\*\* OpenAPI/TS28111\_FaultNrm.yaml \*\*\*

<CODE BEGINS>

openapi: 3.0.1

info:

title: Fault Management NRM

version: 19.2.0

description: >-

OAS 3.0.1 definition of the Fault Supervision MnS

© 2025, 3GPP Organizational Partners (ARIB, ATIS, CCSA, ETSI, TSDSI, TTA, TTC).

All rights reserved.

externalDocs:

description: 3GPP TS 28.111; Fault Management

url: http://www.3gpp.org/ftp/Specs/archive/28\_series/28.111/

servers:

- url: '{MnSRoot}/FaultSupervisionMnS/{MnSversion}'

variables:

MnSRoot:

description: See subclause 4.4.3 of TS 32.158

default: http://example.com/3GPPManagement

MnSversion:

description: Version number of the OpenAPI definition

default: XXX

paths: {}

components:

schemas:

#---- Definition of AlarmRecord ----------------------------------------------------#

AlarmId:

type: string

AlarmType:

type: string

enum:

- 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

- OTHER

readOnly: true

ProbableCause:

description: >-

The value of the probable cause may be a specific standardized string, or any

vendor provided string. Probable cause strings are not standardized in the

present document. They may be added in a future version. Up to then the

mapping of the generic probable cause strings "PROBABLE\_CAUSE\_001" to

"PROBABLE\_CAUSE\_005" is vendor specific.

The value of the probable cause may also be an integer. The mapping of integer

values to probable causes is vendor specific.

oneOf:

- anyOf:

- type: string

enum:

- PROBABLE\_CAUSE\_001

- PROBABLE\_CAUSE\_002

- PROBABLE\_CAUSE\_003

- PROBABLE\_CAUSE\_004

- PROBABLE\_CAUSE\_005

readOnly: true

- type: string

readOnly: true

- type: integer

readOnly: true

SpecificProblem:

oneOf:

- type: string

readOnly: true

- type: integer

readOnly: true

PerceivedSeverity:

type: string

enum:

- INDETERMINATE

- CRITICAL

- MAJOR

- MINOR

- WARNING

- CLEARED

TrendIndication:

type: string

readOnly: true

enum:

- MORE\_SEVERE

- NO\_CHANGE

- LESS\_SEVERE

ThresholdHysteresis:

type: object

required:

- high

properties:

high:

oneOf:

- type: integer

- $ref: 'TS28623\_ComDefs.yaml#/components/schemas/FloatRo'

low:

$ref: 'TS28623\_ComDefs.yaml#/components/schemas/FloatRo'

ThresholdLevelInd:

oneOf:

- type: object

properties:

up:

$ref: '#/components/schemas/ThresholdHysteresis'

- type: object

properties:

down:

$ref: '#/components/schemas/ThresholdHysteresis'

ThresholdCrossing:

type: object

properties:

observedMeasurement:

type: string

readOnly: true

observedValue:

type: number

readOnly: true

thresholdLevel:

$ref: '#/components/schemas/ThresholdLevelInd'

armTime:

$ref: 'TS28623\_ComDefs.yaml#/components/schemas/DateTimeRo'

required:

- observedMeasurement

- observedValue

CorrelatedNotification:

type: object

properties:

sourceObjectInstance:

$ref: 'TS28623\_ComDefs.yaml#/components/schemas/DnRo'

notificationIds:

type: array

items:

$ref: 'TS28623\_ComDefs.yaml#/components/schemas/NotificationId'

required:

- sourceObjectInstance

- notificationIds

CorrelatedNotifications:

type: array

items:

$ref: '#/components/schemas/CorrelatedNotification'

AckState:

type: string

enum:

- ACKNOWLEDGED

- UNACKNOWLEDGED

AlarmRecord:

description: >-

The alarmId is not a property of an alarm record. It is used as key

in the map of alarm records instead.

type: object

properties:

# alarmId:

# $ref: '#/components/schemas/AlarmId'

objectInstance:

$ref: 'TS28623\_ComDefs.yaml#/components/schemas/DnRo'

notificationId:

$ref: 'TS28623\_ComDefs.yaml#/components/schemas/NotificationId'

alarmRaisedTime:

$ref: 'TS28623\_ComDefs.yaml#/components/schemas/DateTimeRo'

alarmChangedTime:

$ref: 'TS28623\_ComDefs.yaml#/components/schemas/DateTimeRo'

alarmClearedTime:

$ref: 'TS28623\_ComDefs.yaml#/components/schemas/DateTimeRo'

alarmType:

$ref: '#/components/schemas/AlarmType'

probableCause:

$ref: '#/components/schemas/ProbableCause'

specificProblem:

$ref: '#/components/schemas/SpecificProblem'

perceivedSeverity:

$ref: '#/components/schemas/PerceivedSeverity'

backedUpStatus:

type: boolean

backUpObject:

$ref: 'TS28623\_ComDefs.yaml#/components/schemas/DnRo'

trendIndication:

$ref: '#/components/schemas/TrendIndication'

thresholdinfo:

$ref: '#/components/schemas/ThresholdCrossing'

correlatedNotifications:

$ref: '#/components/schemas/CorrelatedNotifications'

stateChangeDefinition:

$ref: 'TS28623\_ComDefs.yaml#/components/schemas/AttributeValueChangeSet'

monitoredAttributes:

$ref: 'TS28623\_ComDefs.yaml#/components/schemas/AttributeNameValuePairSet'

proposedRepairActions:

type: string

readOnly: true

additionalText:

type: string

readOnly: true

additionalInformation:

$ref: 'TS28623\_ComDefs.yaml#/components/schemas/AttributeNameValuePairSet'

rootCauseIndicator:

type: boolean

readOnly: true

ackTime:

$ref: 'TS28623\_ComDefs.yaml#/components/schemas/DateTimeRo'

ackUserId:

type: string

ackSystemId:

type: string

ackState:

$ref: '#/components/schemas/AckState'

clearUserId:

type: string

clearSystemId:

type: string

serviceUser:

type: string

readOnly: true

serviceProvider:

type: string

readOnly: true

securityAlarmDetector:

type: string

readOnly: true

clearingType:

type: string

enum:

- MANUAL

- AUTOMATIC

default: AUTOMATIC

AlarmList-Single:

allOf:

- $ref: 'TS28623\_GenericNrm.yaml#/components/schemas/Top'

- type: object

properties:

attributes:

type: object

properties:

administrativeState:

$ref: 'TS28623\_ComDefs.yaml#/components/schemas/AdministrativeState'

operationalState:

$ref: 'TS28623\_ComDefs.yaml#/components/schemas/OperationalState'

numOfAlarmRecords:

type: integer

readOnly: true

lastModification:

$ref: 'TS28623\_ComDefs.yaml#/components/schemas/DateTimeRo'

alarmRecords:

description: >-

This resource represents a map of alarm records.

The alarmIds are used as keys in the map.

type: object

additionalProperties:

$ref: '#/components/schemas/AlarmRecord'

unreliableAlarmScope:

$ref: 'TS28623\_ComDefs.yaml#/components/schemas/DnRo'

#---- Definition of alarm notifications --------------------------------------------#

AlarmNotificationTypes:

type: string

enum:

- notifyNewAlarm

- notifyChangedAlarm

- notifyChangedAlarmGeneral

- notifyAckStateChanged

- notifyCorrelatedNotificationChanged

- notifyComments

- notifyClearedAlarm

- notifyAlarmListRebuilt

- notifyPotentialFaultyAlarmList

AlarmListAlignmentRequirement:

type: string

enum:

- ALIGNMENT\_REQUIRED

- ALIGNMENT\_NOT\_REQUIRED

NotifyNewAlarm:

allOf:

- $ref: 'TS28623\_ComDefs.yaml#/components/schemas/NotificationHeader'

- type: object

required:

- alarmId

- alarmType

- probableCause

- perceivedSeverity

properties:

alarmId:

$ref: '#/components/schemas/AlarmId'

alarmType:

$ref: '#/components/schemas/AlarmType'

probableCause:

$ref: '#/components/schemas/ProbableCause'

specificProblem:

$ref: '#/components/schemas/SpecificProblem'

perceivedSeverity:

$ref: '#/components/schemas/PerceivedSeverity'

backedUpStatus:

type: boolean

backUpObject:

$ref: 'TS28623\_ComDefs.yaml#/components/schemas/Dn'

trendIndication:

$ref: '#/components/schemas/TrendIndication'

thresholdInfo:

$ref: '#/components/schemas/ThresholdCrossing'

correlatedNotifications:

$ref: '#/components/schemas/CorrelatedNotifications'

stateChangeDefinition:

$ref: 'TS28623\_ComDefs.yaml#/components/schemas/AttributeValueChangeSet'

monitoredAttributes:

$ref: 'TS28623\_ComDefs.yaml#/components/schemas/AttributeNameValuePairSet'

proposedRepairActions:

type: string

additionalText:

type: string

additionalInformation:

$ref: 'TS28623\_ComDefs.yaml#/components/schemas/AttributeNameValuePairSet'

rootCauseIndicator:

type: boolean

clearingType:

type: string

enum:

- MANUAL

- AUTOMATIC

NotifyNewSecAlarm:

allOf:

- $ref: 'TS28623\_ComDefs.yaml#/components/schemas/NotificationHeader'

- type: object

required:

- alarmId

- alarmType

- probableCause

- perceivedSeverity

- serviceUser

- serviceProvider

- securityAlarmDetector

properties:

alarmId:

$ref: '#/components/schemas/AlarmId'

alarmType:

$ref: '#/components/schemas/AlarmType'

probableCause:

$ref: '#/components/schemas/ProbableCause'

specificProblem:

$ref: '#/components/schemas/SpecificProblem'

perceivedSeverity:

$ref: '#/components/schemas/PerceivedSeverity'

correlatedNotifications:

$ref: '#/components/schemas/CorrelatedNotifications'

additionalText:

type: string

additionalInformation:

$ref: 'TS28623\_ComDefs.yaml#/components/schemas/AttributeNameValuePairSet'

rootCauseIndicator:

type: boolean

serviceUser:

type: string

serviceProvider:

type: string

securityAlarmDetector:

type: string

clearingType:

type: string

enum:

- MANUAL

- AUTOMATIC

NotifyClearedAlarm:

allOf:

- $ref: 'TS28623\_ComDefs.yaml#/components/schemas/NotificationHeader'

- type: object

required:

- alarmId

- alarmType

- probableCause

- perceivedSeverity

properties:

alarmId:

$ref: '#/components/schemas/AlarmId'

alarmType:

$ref: '#/components/schemas/AlarmType'

probableCause:

$ref: '#/components/schemas/ProbableCause'

specificProblem:

$ref: '#/components/schemas/SpecificProblem'

perceivedSeverity:

$ref: '#/components/schemas/PerceivedSeverity'

correlatedNotifications:

$ref: '#/components/schemas/CorrelatedNotifications'

clearUserId:

type: string

clearSystemId:

type: string

NotifyChangedAlarm:

allOf:

- $ref: 'TS28623\_ComDefs.yaml#/components/schemas/NotificationHeader'

- type: object

required:

- alarmId

- alarmType

- probableCause

- perceivedSeverity

properties:

alarmId:

$ref: '#/components/schemas/AlarmId'

alarmType:

$ref: '#/components/schemas/AlarmType'

probableCause:

$ref: '#/components/schemas/ProbableCause'

specificProblem:

$ref: '#/components/schemas/SpecificProblem'

perceivedSeverity:

$ref: '#/components/schemas/PerceivedSeverity'

NotifyChangedAlarmGeneral:

allOf:

- $ref: 'TS28623\_ComDefs.yaml#/components/schemas/NotificationHeader'

- type: object

required:

- alarmId

- alarmType

- probableCause

properties:

alarmId:

$ref: '#/components/schemas/AlarmId'

alarmType:

$ref: '#/components/schemas/AlarmType'

probableCause:

$ref: '#/components/schemas/ProbableCause'

specificProblem:

$ref: '#/components/schemas/SpecificProblem'

perceivedSeverity:

$ref: '#/components/schemas/PerceivedSeverity'

correlatedNotifications:

$ref: '#/components/schemas/CorrelatedNotifications'

backedUpStatus:

type: boolean

backUpObject:

$ref: 'TS28623\_ComDefs.yaml#/components/schemas/Dn'

trendIndication:

$ref: '#/components/schemas/TrendIndication'

thresholdInfo:

$ref: '#/components/schemas/ThresholdCrossing'

stateChangeDefinition:

$ref: 'TS28623\_ComDefs.yaml#/components/schemas/AttributeValueChangeSet'

monitoredAttributes:

$ref: 'TS28623\_ComDefs.yaml#/components/schemas/AttributeNameValuePairSet'

proposedRepairActions:

type: string

additionalText:

type: string

additionalInformation:

$ref: 'TS28623\_ComDefs.yaml#/components/schemas/AttributeNameValuePairSet'

rootCauseIndicator:

type: boolean

changedAlarmAttributes:

$ref: 'TS28623\_ComDefs.yaml#/components/schemas/AttributeNameValuePairSet'

NotifyChangedSecAlarmGeneral:

allOf:

- $ref: 'TS28623\_ComDefs.yaml#/components/schemas/NotificationHeader'

- type: object

required:

- alarmId

- alarmType

- probableCause

- serviceUser

- serviceProvider

- securityAlarmDetector

properties:

alarmId:

$ref: '#/components/schemas/AlarmId'

alarmType:

$ref: '#/components/schemas/AlarmType'

probableCause:

$ref: '#/components/schemas/ProbableCause'

specificProblem:

$ref: '#/components/schemas/SpecificProblem'

perceivedSeverity:

$ref: '#/components/schemas/PerceivedSeverity'

correlatedNotifications:

$ref: '#/components/schemas/CorrelatedNotifications'

additionalText:

type: string

additionalInformation:

$ref: 'TS28623\_ComDefs.yaml#/components/schemas/AttributeNameValuePairSet'

rootCauseIndicator:

type: boolean

serviceUser:

type: string

serviceProvider:

type: string

securityAlarmDetector:

type: string

changedAlarmAttributes:

$ref: 'TS28623\_ComDefs.yaml#/components/schemas/AttributeNameValuePairSet'

NotifyCorrelatedNotificationChanged:

allOf:

- $ref: 'TS28623\_ComDefs.yaml#/components/schemas/NotificationHeader'

- type: object

required:

- alarmId

- correlatedNotifications

- alarmType

- probableCause

properties:

alarmId:

$ref: '#/components/schemas/AlarmId'

alarmType:

$ref: '#/components/schemas/AlarmType'

probableCause:

$ref: '#/components/schemas/ProbableCause'

specificProblem:

$ref: '#/components/schemas/SpecificProblem'

correlatedNotifications:

$ref: '#/components/schemas/CorrelatedNotifications'

rootCauseIndicator:

type: boolean

NotifyAckStateChanged:

allOf:

- $ref: 'TS28623\_ComDefs.yaml#/components/schemas/NotificationHeader'

- type: object

required:

- alarmId

- alarmType

- probableCause

- perceivedSeverity

- ackState

- ackUserId

properties:

alarmId:

$ref: '#/components/schemas/AlarmId'

alarmType:

$ref: '#/components/schemas/AlarmType'

probableCause:

$ref: '#/components/schemas/ProbableCause'

specificProblem:

$ref: '#/components/schemas/SpecificProblem'

perceivedSeverity:

$ref: '#/components/schemas/PerceivedSeverity'

ackState:

$ref: '#/components/schemas/AckState'

ackUserId:

type: string

ackSystemId:

type: string

NotifyComments:

allOf:

- $ref: 'TS28623\_ComDefs.yaml#/components/schemas/NotificationHeader'

- type: object

required:

- alarmId

- alarmType

- probableCause

- perceivedSeverity

- comments

properties:

alarmId:

$ref: '#/components/schemas/AlarmId'

alarmType:

$ref: '#/components/schemas/AlarmType'

probableCause:

$ref: '#/components/schemas/ProbableCause'

specificProblem:

$ref: '#/components/schemas/SpecificProblem'

perceivedSeverity:

$ref: '#/components/schemas/PerceivedSeverity'

comments:

$ref: '#/components/schemas/Comments'

NotifyPotentialFaultyAlarmList:

allOf:

- $ref: 'TS28623\_ComDefs.yaml#/components/schemas/NotificationHeader'

- type: object

required:

- reason

properties:

reason:

type: string

NotifyAlarmListRebuilt:

allOf:

- $ref: 'TS28623\_ComDefs.yaml#/components/schemas/NotificationHeader'

- type: object

required:

- reason

properties:

reason:

type: string

alarmListAlignmentRequirement:

$ref: '#/components/schemas/AlarmListAlignmentRequirement'

#---- Definition of resources ------------------------------------------------------#

Comment:

type: object

properties:

commentTime:

$ref: 'TS28623\_ComDefs.yaml#/components/schemas/DateTimeRo'

commentUserId:

type: string

commentSystemId:

type: string

commentText:

type: string

Comments:

description: >-

Collection of comments. The comment identifiers are allocated by the

MnS producer and used as key in the map.

type: object

additionalProperties:

$ref: '#/components/schemas/Comment'

#----- Definitions in TS 28.111 for TS 28.532 --------------------------#

resources-faultNrm:

oneOf:

- $ref: '#/components/schemas/AlarmList-Single'

#----- Definitions in TS 28.111 for TS 28.532 --------------------------#

<CODE ENDS>

\*\*\* END OF CHANGE 1 \*\*\*