**3GPP TSG- Meeting #**

**, , - revision S5-241131**

|  |
| --- |
| *CR-Form-v12.3* |
| **CHANGE REQUEST** |
|  |
|  |  | **CR** |  | **rev** |  | **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:*** |  |  | ***Date:*** |  |
|  |  |  |  |  |
| ***Category:*** |  |  | ***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 canbe 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:*** | Some IOC can not be created by the MnS consumer only by the producer. Just as other IOC properties this need to be visible in the YANG modules. |
|  |  |
| ***Summary of change:*** | A new YANG extension only-system-created is added to provide the missing information. (Only YANG updates.) |
|  |  |
| ***Consequences if not approved:*** | The consumer cannot create specific IOCs/MOIs and will not receive information about the failure reason. |
|  |  |
| ***Clauses affected:*** | Forge impact only. |
|  |  |
|  | **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 link: <https://forge.3gpp.org/rep/sa5/MnS/-/merge_requests/1063> at commit 5b748eb38303639ca0fce23b0897d82ba7a55e46 |
|  |  |
| ***This CR's revision history:*** |  |
|  |  |

Forge MR link: <https://forge.3gpp.org/rep/sa5/MnS/-/merge_requests/1063> at commit 09d00313a7bb22262f6ba38116953fd52b94c005

\*\*\* 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 2024, 3GPP Organizational Partners (ARIB, ATIS, CCSA, ETSI, TSDSI,

 TTA, TTC). All rights reserved.";

 reference "3GPP TS 28.623

 Generic Network Resource Model (NRM)

 Integration Reference Point (IRP);

 Solution Set (SS) definitions

 3GPP TS 28.622

 Generic Network Resource Model (NRM)

 Integration Reference Point (IRP);

 Information Service (IS)";

 revision 2024-04-05 { reference CR-0332 ; }

 revision 2023-09-17 { reference CR-0270 ; }

 revision 2023-05-09 { reference CR-0249; }

 revision 2022-10-31 { reference CR-0195; }

 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 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 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;

 }

 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;

 }

 grouping ThresholdInfoGrp {

 leaf measurementType {

 type string;

 mandatory true;

 }

 leaf direction {

 type enumeration {

 enum INCREASING;

 enum DECREASING;

 }

 mandatory true;

 description "

 If it is 'Increasing', the threshold crossing notification is

 triggered when the measurement value equals or exceeds a

 thresholdValue.

 If it is 'Decreasing', the threshold crossing notification is

 triggered when the measurement value equals or below a

 thresholdValue.";

 }

 leaf thresholdLevel {

 type string;

 }

 leaf thresholdValue {

 type string;

 }

 leaf hysteresis {

 type string;

 description "The hysteresis has a threshold high and a threshold

 low value that are different from the threshold value.

 A hysteresis, therefore, defines the threshold-high and

 threshold-low levels within which the measurementType value is

 allowed to oscillate without triggering the threshold crossing

 notification.";

 }

 }

 list thresholdInfo {

 config false ;

 yext3gpp:notNotifyable;

 description "Indicates the crossed threshold";

 uses ThresholdInfoGrp;

 }

 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;

 }

 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 notificationId {

 type int32;

 min-elements 1;

 }

 }

 }

 grouping AlarmListGrp {

 description "Represents the AlarmList IOC.";

 leaf administrativeState {

 type types3gpp:AdministrativeState ;

 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;

 }

 }

 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 cannot 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";

 uses top3gpp:Top\_Grp ;

 container attributes {

 uses AlarmListGrp ;

 }

 }

 }

}

<CODE ENDS>

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

\*\*\* START OF CHANGE 2 \*\*\*

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

<CODE BEGINS>

module \_3gpp-common-yang-extensions {

 yang-version 1.1;

 namespace urn:3gpp:sa5:\_3gpp-common-yang-extensions ;

 prefix yext3gpp ;

 organization "3GPP SA5";

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

 description "The module defines YANG extensions needed

 3GPP YANG modeling.

 Extensions MUST be defined with the following structure in the

 description statement:

 - What is this statement.

 - Newline,

 - This statement can be a substatement of the xxx statements with

 cardinality x..y.

 - This statement can have the following substatements with

 cardinality x..y.

 - Newline

 - Is changing this statement an editorial, BC(backwards compatible)

 or NBC(non-BC) change?

 - Newline.

 - The argument its meaning and type. Preferably use YANG types and

 constraints to define the argument's type.

 Any extension statement can be added with a

 deviation/deviate add statement. In this case the restriction about

 the parent statement of the extension SHALL be evaluated based on the

 target of the deviation statement.

 Support for this module does not mean that a YANG server implements

 support for each of these extensions.

 Implementers of each specific module using an extensions MUST check

 if the server implements support for the used extension.

 Note: modules use many extensions which individual

 implementations MAY or MAY NOT support.

 If support for an extension is missing the extension statement needs

 individual handling or it SHOULD be removed from the module using

 the extension e.g. with a deviation.

 Copyright 2024, 3GPP Organizational Partners (ARIB, ATIS, CCSA, ETSI, TSDSI,

 TTA, TTC). All rights reserved.";

 reference "3GPP TS 28.623

 Generic Network Resource Model (NRM)

 Integration Reference Point (IRP);

 Solution Set (SS) definitions

 3GPP TS 28.623";

 revision 2024-04-05 { reference "CR-0332, 0333" ; }

 revision 2023-09-17 { reference "CR-0270, 0271" ; }

 revision 2022-10-31 { reference "CR-0195, 0196"; }

 revision 2019-06-23 { reference "Initial version"; }

 extension only-system-created {

 description

 "Indicates that the MOI can only be created and deleted by the

 MnS producer.

 System created MOIs can have both read-only and read-write data

 node children. Implementers shall add the extension to standard

 models if they restrict creation/deletion of an IOC.

 The statement MUST only be a sub-statement of the list statement,

 with the parent list statement representing an IOC.

 Zero or one only-system-created statement per parent statement is

 allowed. No sub-statements are allowed.

 Adding this statement is an NBC change, removing it is BC.";

 }

 extension notNotifyable {

 description

 "Indicates that data change notifications shall not be sent

 for this attribute. If the extension is not present and other

 conditions are fulfilled data change notification should be sent.

 If a list or container already has the notNotifyable

 extension, that is also valid for all contained data nodes.

 The statement MUST only be a substatement of a leaf, leaf-list, list,

 container statement that is contained within the 'attributes'

 container of an IOC and that represents an attribute or sub-parts of

 an attribute .

 Zero or one notNotifyable statement is allowed per parent statement.

 NO substatements are allowed.

 Adding this statement is an NBC change, removing it is BC.";

 }

 extension inVariant {

 description

 "Indicates that the value for the data node can only be set when the list

 data node representing the containing object(MOI) is created.

 The above statement in YANG terms means, that the value for the data

 node can only be set when the list entry that is the parent of the

 restricted node's ancestor container named 'attributes' is being created.

 To change the value after that, the mentioned

 list data node must be deleted and recreated with the restricted data node

 having the new value.

 If a list or container already has the inVariant

 extension, that is also valid for all contained data nodes.

 It is unnecessary to use and MUST NOT be used for key leafs.

 The statement MUST only be a substatement of a leaf, leaf-list, list

 statement that is config=true.

 Zero or one inVariant statement is allowed per parent statement.

 NO substatements are allowed.

 Adding this statement is an NBC change, removing it is BC.";

 }

 extension initial-value {

 description "Specifies a value that the system will set for a leaf

 leaf-list if a value is not specified for it when its parent list

 or container is created. The value has no effect in any other

 modification e.g. changing or removing the value.

 The description statement of the parent statement SHOULD contain

 the label 'Initial-value: ' followed by the text from the argument.

 The statement MUST only be a substatement of a leaf or leaf-list.

 The statement MUST NOT be present if the leaf or the leaf-list

 has a default statement or the type used for the data node

 has a default value.

 The statement MUST NOT be used for config=false data or in an

 action, rpc or notification.

 Zero or one initial-value statements are allowed for a leaf parent

 statement. Zero or more initial-value statements are allowed for a

 leaf-list parent statement. If the leaf-list is ordered-by user, the

 initial values are stored in the order they appear in the YANG definition.

 NO substatements are allowed.

 Always consider using a YANG-default statement instead.

 Modification of the initial-value is a non-backwards-compatible change.

 The argument specifies a single initial value for a leaf or leaf-list.

 The value MUST be part of the valuespace of the leaf/leaf-list.

 It follows the same rules as the argument of the default statement.";

 argument "initial-value";

 }

}

<CODE ENDS>

\*\*\* END OF CHANGE 2 \*\*\*