3GPP TSG-SA5 Meeting #136-e S5-212376

**, , -**

|  |
| --- |
| *CR-Form-v12.1* |
| **CHANGE REQUEST** |
|  |
|  | **28.623** | **CR** | **crno** | **rev** | **-** | **Current version:** | **16.6.0** |  |
|  |
| *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:***  | YANG update |
|  |  |
| ***Source to WG:*** | Ericsson Hungary Ltd. |
| ***Source to TSG:*** | S5 |
|  |  |
| ***Work item code:*** | eNRM |  | ***Date:*** | 2021-03-09 |
|  |  |  |  |  |
| ***Category:*** | **F** |  | ***Release:*** | Rel-16 |
|  | *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-15 (Release 15)Rel-16 (Release 16)Rel-17 (Release 17)Rel-18 (Release 18)* |
|  |  |
| ***Reason for change:*** | Correcting YANG errors |
|  |  |
| ***Summary of change:*** | Update YANG code to follow stage 2 definitions and correct earlier YANG compilation errors. |
|  |  |
| ***Consequences if not approved:*** | YANG code does not compile and/or does not follow the stage 2 definitions |
|  |  |
| ***Clauses affected:*** | D.2.2, D.2.3, D.2.5, D.2.26a, D.2.10 |
|  |  |
|  | **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:*** | The CR is based on 211351 and 212287<https://forge.3gpp.org/rep/sa5/MnS/tree/S5-211299_Rel-16_CR_28.623_Correct_NtfSubscriptionControl_containment_in_YANG> |
|  |  |
| ***This CR's revision history:*** |  |

***First change***

## D.2.2 module \_3gpp-common-managed-element.yang

<CODE BEGINS>

module \_3gpp-common-managed-element {

 yang-version 1.1;

 namespace urn:3gpp:sa5:\_3gpp-common-managed-element;

 prefix "me3gpp";

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

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

 import \_3gpp-common-measurements { prefix meas3gpp; }

 import \_3gpp-common-subscription-control { prefix subscr3gpp; }

 import \_3gpp-common-fm { prefix fm3gpp; }

 import \_3gpp-common-trace { prefix trace3gpp; }

 organization "3GPP SA5";

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

 description "Defines ManagedElement which will be augmented

 by other IOCs";

 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)

 3GPP TS 28.620

 Umbrella Information Model (UIM)";

 revision 2021-01-16 { reference "CR-0120"; }

 revision 2020-08-06 { reference "CR-0102"; }

 revision 2020-08-03 { reference "CR-0095"; }

 revision 2020-06-08 { reference "CR-0092"; }

 revision 2020-05-12 { reference "CR0084"; }

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

 revision 2019-06-17 { reference " S5-203316"; }

 revision 2019-05-08 { reference "Initial revision"; }

 feature MeasurementsUnderManagedElement {

 description "The MeasurementSubtree shall be contained under

 ManagedElement";

 }

 feature SubscriptionControlUnderManagedElement {

 description "The SubscriptionControlSubtree shall be contained under

 ManagedElement";

 }

 feature FmUnderManagedElement {

 description "The FmSubtree shall be contained under ManagedElement";

 }

 feature TraceUnderManagedElement {

 description "The TraceSubtree shall be contained under ManageElement";

 }

 feature DESManagementFunction {

 description "Class representing Distributed SON or Domain-Centralized SON

 Energy Saving feature. The DESManagementFunction shall be contained under

 ManagedElement.";

 }

 feature DMROFunction {

 description "Class representing D-SON function of MRO feature. The

 DMROFunction shall be contained under ManagedElement.";

 }

 feature DRACHOptimizationFunction {

 description "Class representing D-SON function of RACH optimization

 feature. The DRACHOptimizationFunction shall be contained under

 ManagedElement.";

 }

 feature DPCIConfigurationFunction {

 description "Class representing Distributed SON or Domain-Centralized SON

 function of PCI configuration feature. The DPCIConfigurationFunction shall

 be contained under ManagedElement.";

 }

 feature CPCIConfigurationFunction {

 description "Class representing Cross Domain-Centralized SON function of PCI

 configuration feature. The CPCIConfigurationFunction shall be contained

 under ManagedElement.";

 }

 feature CESManagementFunction {

 description "Class representing Cross Domain-Centralized SON Energy Saving

 feature. The CESManagementFunction shall be contained under

 ManagedElement.";

 }

 grouping ManagedElement\_Grp {

 description "Abstract class representing telecommunications resources.

 An ME communicates with a manager (directly or indirectly) for the

 purpose of being monitored and/or controlled. MEs may perform element

 management functionality.

 An ME (and its contained Function\_(s)) may or may not be geographically

 distributed. An ME (and its contained Function\_(s)) is often referred

 to as a Network Element";

 leaf dnPrefix {

 description "Provides naming context that allows the Managed

 Elements to be partitioned into logical domains.

 A Distingushed Name(DN) is defined by 3GPP TS 32.300,

 which splits the DN into a DN Prefix and Local DN";

 type types3gpp:DistinguishedName;

 }

 leaf userLabel {

 description "A user-friendly (and user assignable) name of this object.";

 type string;

 }

 leaf locationName {

 description "The physical location (e.g. an address) of an entity

 represented by a (derivative of) ManagedElement\_. It may contain no

 information to support the case where the derivative of

 ManagedElement\_ needs to represent a distributed multi-location NE.";

 config false;

 type string;

 }

 leaf-list managedBy {

 description "Relates to the role played by ManagementSystem\_ in the

 between ManagedSystem\_ and ManagedElement\_. This attribute contains

 a list of the DN(s) of the related subclasses of

 ManagementSystem\_ instance(s).";

 config false;

 type types3gpp:DistinguishedName;

 }

 leaf-list managedElementTypeList {

 description "The type of functionality provided by the ManagedElement.

 It may represent one ME functionality or a combination of

 more than one functionality.

 1) The allowed values of this attribute are the names of the IOC(s)

 that are (a) derived/subclassed from ManagedFunction and (b) directly

 name-contained by ManagedElement IOC (on the first level below

 ManagedElement), but with the string ’Function’ excluded.

 2) If a ManagedElement contains multiple instances of a ManagedFunction

 this attribute will not contain repeated values.

 3) The capitalisation (usage of upper/lower case) of characters in this

 attribute is insignificant. Thus, the NodeB should be case insensitive

 when reading these values.

 4) Two examples of allowed values are:

 - NodeB;

 - HLR, VLR.";

 config false;

 min-elements 1;

 type string;

 }

 }

 grouping ManagedElementGrp {

 description "Represents telecommunications equipment or

 TMN entities within the telecommunications network providing support

 and/or service to the subscriber.";

 uses ManagedElement\_Grp;

 uses meas3gpp:SupportedPerfMetricGroupGrp;

 leaf vendorName {

 config false;

 type string;

 }

 leaf userDefinedState {

 type string;

 description "An operator defined state for operator specific usage";

 }

 leaf swVersion {

 config false;

 type string;

 }

 leaf priorityLabel {

 type uint32;

 mandatory true;

 }

 }

 list ManagedElement {

 description "Represents telecommunications equipment or TMN entities within

 the telecommunications network providing support and/or service to the

 subscriber. An ManagedElement IOC is used to represent a Network Element

 defined in TS 32.101 including virtualizeation or non-virtualization

 scenario. An ManagedElement instance is used for communicating with a

 manager (directly or indirectly) over one or more management interfaces

 for the purpose of being monitored and/or controlled. ManagedElement may

 or may not additionally perform element management functionality.

 An ManagedElement contains equipment that may or may not be geographically

 distributed.

 A telecommunication equipment has software and hardware components. The

 ManagedElement IOC described above represents following two case:

 - In the case when the software component is designed to run on dedicated

 hardware component, the ManagedElement IOC description includes both

 software and hardware components.

 - In the case when the software is designed to run on ETSI NFV defined

 NFVI [15], the ManagedElement IOC description would exclude the NFVI

 component supporting the above mentioned subject software.

 A ManagedElement may be contained in either a SubNetwork or in a MeContext

 instance. A single ManagedElement may also exist stand-alone with no

 parent at all.

 The relation of ManagedElement IOC and ManagedFunction IOC can be

 described as following：

 - A ManaagedElement instance may have 1..1 containment relationship to a

 ManagedFunction instance. In this case, the ManagedElement IOC may be

 used to represent a NE with single functionality. For example, a

 ManagedElement is used to represent the 3GPP defined RNC node;

 - A ManagedElement instance may have 1..N containment relationship to

 multiple ManagedFunction IOC instances. In this case, the ManagedElement

 IOC may be used to represent a NE with combined ManagedFunction

 funcationality (as indicated by the managedElementType attribute and the

 contained instances of different ManagedFunction IOCs).For example, a

 ManagedElement is used to represent the combined functionality of 3GPP

 defined gNBCUCPFuntion, gNBCUUPFunction and gNBDUFunction";

 key id;

 uses top3gpp:Top\_Grp;

 container attributes {

 uses ManagedElementGrp;

 }

 uses meas3gpp:MeasurementSubtree {

 if-feature MeasurementsUnderManagedElement;

 }

 uses subscr3gpp:SubscriptionControlSubtree {

 if-feature SubscriptionControlUnderManagedElement;

 }

 uses fm3gpp:FmSubtree {

 if-feature FmUnderManagedElement;

 }

 uses trace3gpp:TraceSubtree {

 if-feature TraceUnderManagedElement;

 }

 }

}

<CODE ENDS>

## D.2.3 module \_3gpp-common-managed-function.yang

<CODE BEGINS>

module \_3gpp-common-managed-function {

 yang-version 1.1;

 namespace urn:3gpp:sa5:\_3gpp-common-managed-function;

 prefix mf3gpp;

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

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

 import \_3gpp-common-measurements { prefix meas3gpp; }

 import \_3gpp-common-trace { prefix trace3gpp; }

 organization "3GPP SA5";

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

 description "The module defines a base class/grouping for major 3GPP

 functions.";

 reference

 "3GPP TS 28.622

 Generic Network Resource Model (NRM)

 Integration Reference Point (IRP);

 Information Service (IS)

 3GPP TS 28.620

 Umbrella Information Model (UIM)";

 revision 2021-01-25 { reference "CR-0122"; }

 revision 2020-09-30 { reference "CR-bbbb"; }

 revision 2020-08-06 { reference "CR-0102"; }

 revision 2020-08-03 { reference "CR-0095"; }

 revision 2020-06-23 { reference "CR-085"; }

 revision 2020-06-08 { reference "CR-0092"; }

 revision 2019-11-21 { reference "S5-197275, S5-197735"; }

 revision 2019-10-28 { reference S5-193518 ; }

 revision 2019-06-18 { reference "Initial revision"; }

 feature MeasurementsUnderManagedFunction {

 description "The MeasurementSubtree shall be contained under ManageElement";

 }

 feature TraceUnderManagedFunction {

 description "The TraceSubtree shall be contained under ManagedFunction";

 }

 grouping Operation {

 description "This data type represents an Operation.";

 reference "3gpp TS 28.622";

 leaf name {

 type string;

 mandatory true;

 }

 leaf-list allowedNFTypes {

 type string;

 min-elements 1;

 description "The type of the managed NF service instance

 The specifc values allowed are described in TS 23.501";

 }

 leaf operationSemantics {

 type enumeration {

 enum REQUEST\_RESPONSE;

 enum SUBSCRIBE\_NOTIFY;

 }

 config false;

 mandatory true;

 description "Semantics type of the operation.";

 reference "3GPP TS 23.502";

 }

 }

 grouping ManagedNFServiceGrp {

 description "A ManagedNFService represents a Network Function (NF) service.";

 reference "Clause 7 of 3GPP TS 23.501.";

 leaf userLabel {

 type string;

 description "A user-friendly (and user assignable) name of this object.";

 }

 leaf nFServiceType {

 config false;

 mandatory true;

 type string;

 description "The type of the managed NF service instance

 The specifc values allowed are described in clause 7.2 of TS 23.501";

 }

 list sAP {

 key "host port";

 min-elements 1;

 max-elements 1;

 description "The service access point of the managed NF service instance";

 uses types3gpp:SAP;

 }

 list operations {

 key name;

 min-elements 1;

 uses Operation ;

 description "Set of operations supported by the managed NF

 service instance";

 }

 leaf administrativeState {

 type types3gpp:AdministrativeState;

 mandatory true;

 description "Permission to use or prohibition against using the instance";

 }

 leaf operationalState {

 type types3gpp:OperationalState;

 config false;

 mandatory true;

 description "Describes whether the resource is installed and working";

 }

 leaf usageState {

 type types3gpp:usageState ;

 config false;

 mandatory true;

 description "Describes whether the resource is actively in use at a

 specific instant, and if so, whether or not it has spare

 capacity for additional users.";

 }

 leaf registrationState {

 type enumeration {

 enum REGISTERED;

 enum DEREGISTERED;

 }

 config false;

}

 }

 grouping Function\_Grp {

 description "A base grouping for 3GPP functions.";

 leaf userLabel {

 type string;

 description "A user-friendly (and user assignable) name of this object.";

 }

 }

 grouping ManagedFunctionGrp {

 description "Abstract root class to be inherited/reused by classes

 representing 3GPP functions.

 Anywhere this grouping is used by classes inheriting from ManagedFunction

 the list representing the inheriting class needs to include all

 contained classes of ManagedFunction too. Contained classes are

 either

 - augmented into the Function class or

 - shall be included in the list representing the inheriting class

 using the grouping ManagedFunctionContainedClasses:

 1) EP\_RP solved using augment

 2) uses mf3gpp:ManagedFunctionContainedClasses;

 ";

 uses Function\_Grp;

 list vnfParametersList {

 key vnfInstanceId;

 description "Contains the parameter set of the VNF

 instance(s) corresponding to an NE.

 The presence of this list indicates that the ManagedFunction

 represented is realized by one or more VNF instance(s). Otherwise it

 shall be absent.

 The presence of a vnfParametersList entry, whose vnfInstanceId with a

 string length of zero, in createMO operation can trigger the

 instantiation of the related VNF/VNFC instances.";

 leaf vnfInstanceId {

 type string ;

 description "VNF instance identifier";

 reference "ETSI GS NFV-IFA 008 v2.1.1:

 Network Functions Virtualisation (NFV); Management and Orchestration;

 Ve-Vnfm reference point - Interface and Information Model Specification

 section 9.4.2

 ETSI GS NFV-IFA 015 v2.1.2: Network Functions Virtualisation (NFV);

 Management and Orchestration; Report on NFV Information Model

 section B2.4.2.1.2.3";

 }

 leaf vnfdId {

 type string ;

 description "Identifier of the VNFD on which the VNF instance is based.

 The absence of the leaf or a string length of zero for vnfInstanceId

 means the VNF instance(s) does not exist (e.g. has not been

 instantiated yet, has already been terminated).";

 reference "ETSI GS NFV-IFA 008 v2.1.1:

 Network Functions Virtualisation (NFV); Management and Orchestration;

 Ve-Vnfm reference point - Interface and Information Model Specification

 section 9.4.2";

 }

 leaf flavourId {

 type string ;

 description "Identifier of the VNF Deployment Flavour applied to this

 VNF instance.";

 reference "ETSI GS NFV-IFA 008 v2.1.1:

 Network Functions Virtualisation (NFV) Management and Orchestration";

 }

 leaf autoScalable {

 type boolean ;

 mandatory true;

 description "Indicator of whether the auto-scaling of this

 VNF instance is enabled or disabled.";

 }

 }

 list peeParametersList {

 key idx;

 description "Contains the parameter set for the control

 and monitoring of power, energy and environmental parameters of

 ManagedFunction instance(s).";

 leaf idx { type uint32; }

 leaf siteIdentification {

 type string;

 mandatory true;

 description "The identification of the site where the

 ManagedFunction resides.";

 }

 leaf siteLatitude {

 type decimal64 {

 fraction-digits 4;

 range "-90.0000..+90.0000";

 }

 description "The latitude of the site where the ManagedFunction

 instance resides, based on World Geodetic System (1984 version)

 global reference frame (WGS 84). Positive values correspond to

 the northern hemisphere. This attribute is optional in case of

 BTSFunction and RNCFunction instance(s).";

 }

 leaf siteLongitude {

 type decimal64 {

 fraction-digits 4;

 range "-180.0000..+180.0000";

 }

 description "The longitude of the site where the ManagedFunction

 instance resides, based on World Geodetic System (1984 version)

 global reference frame (WGS 84). Positive values correspond to

 degrees east of 0 degrees longitude. This attribute is optional in

 case of BTSFunction and RNCFunction instance(s).";

 }

 leaf siteDescription {

 type string;

 mandatory true;

 description "An operator defined description of the site where

 the ManagedFunction instance resides.";

 }

 leaf equipmentType {

 type string;

 mandatory true;

 description "The type of equipment where the managedFunction

 instance resides.";

 reference "clause 4.4.1 of ETSI ES 202 336-12";

 }

 leaf environmentType {

 type string;

 mandatory true;

 description "The type of environment where the managedFunction

 instance resides.";

 reference "clause 4.4.1 of ETSI ES 202 336-12";

 }

 leaf powerInterface {

 type string;

 mandatory true;

 description "The type of power.";

 reference "clause 4.4.1 of ETSI ES 202 336-12";

 }

 }

 leaf priorityLabel {

 mandatory true;

 type uint32;

 }

 uses meas3gpp:SupportedPerfMetricGroupGrp;

 }

 grouping ManagedFunctionContainedClasses {

 description "A grouping used to containe classes (lists) contained by

 the abstract IOC ManagedFunction";

 list ManagedNFService {

 description "Represents a Network Function (NF)";

 reference "3GPP TS 23.501";

 key id;

 uses top3gpp:Top\_Grp;

 container attributes {

 uses ManagedNFServiceGrp;

 }

 }

 uses meas3gpp:MeasurementSubtree {

 if-feature MeasurementsUnderManagedFunction ;

 }

 uses trace3gpp:TraceSubtree {

 if-feature TraceUnderManagedFunction ;

 }

 }

}

<CODE ENDS>

***Next change***

## D.2.5 module \_3gpp-common-subnetwork.yang

<CODE BEGINS>

module \_3gpp-common-subnetwork {

 yang-version 1.1;

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

 prefix "subnet3gpp";

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

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

 import \_3gpp-common-measurements { prefix meas3gpp; }

 import \_3gpp-common-subscription-control { prefix subscr3gpp; }

 import \_3gpp-common-fm { prefix fm3gpp; }

 import \_3gpp-common-trace { prefix trace3gpp; }

 import ietf-yang-schema-mount { prefix yangmnt; }

 organization "3GPP SA5";

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

 description "Defines basic SubNetwork which will be augmented by other IOCs";

 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)

 3GPP TS 28.620

 Umbrella Information Model (UIM)";

 revision 2021-01-16 { reference "CR-0120"; }

 revision 2020-08-06 { reference "CR-0102"; }

 revision 2020-06-08 { reference "CR-0092"; }

 revision 2020-05-08 {

 reference "S5-203316";

 }

 revision 2020-03-11 {

 description "Added KPIs and corrections";

 reference "S5-201365, S5-201581, SP-200229";

 }

 revision 2020-02-24 {

 reference "S5-201365";

 }

 revision 2019-06-17 {

 reference "Initial revision";

 }

 feature ExternalsUnderSubNetwork {

 description "Classes representing external entities like EUtranFrequency,

 ExternalGNBCUCPFunction, ExternalENBFunction

 are contained under a Subnetwork list/class.";

 }

 feature MeasurementsUnderSubNetwork {

 description "The MeasurementSubtree shall be contained under SubNetwork";

 }

 feature SubscriptionControlUnderSubNetwork {

 description "The SubscriptionControlSubtree shall be contained under

 SubNetwork";

 }

 feature FmUnderSubNetwork {

 description "The FmSubtree shall be contained under SubNetwork";

 }

 feature TraceUnderSubNetwork {

 description "The TraceSubtree shall be contained under SubNetwork";

 }

 feature DESManagementFunction {

 description "Class representing Distributed SON or Domain-Centralized SON

 Energy Saving feature. The DESManagementFunction shall be contained under

 SubNetwork.";

 }

 feature DMROFunction {

 description "Class representing D-SON function of MRO feature. The

 DMROFunction shall be contained under SubNetwork.";

 }

 feature DRACHOptimizationFunction {

 description "Class representing D-SON function of RACH optimization feature.

 The DRACHOptimizationFunction shall be contained under SubNetwork.";

 }

 feature DPCIConfigurationFunction {

 description "Class representing Distributed SON or Domain-Centralized SON

 function of PCI configuration feature. The DPCIConfigurationFunction shall

 be contained under SubNetwork.";

 }

 feature CPCIConfigurationFunction {

 description "Class representing Cross Domain-Centralized SON function of PCI

 configuration feature. The CPCIConfigurationFunction shall be contained

 under SubNetwork.";

 }

 feature CESManagementFunction {

 description "Class representing Cross Domain-Centralized SON Energy Saving

 feature. The CESManagementFunction shall be contained under SubNetwork.";

 }

 grouping Domain\_Grp {

 description "A domain is a partition of instances of managed entities

 such that :

 - the group represents a topological structure which describes the

 potential for connectivity

 - Subject to common administration

 - With common characteristics";

 leaf dnPrefix {

 type types3gpp:DistinguishedName;

 reference "Annex C of 32.300 ";

 }

 leaf userLabel {

 type string;

 description "A user-friendly (and user assignable) name of this object.";

 }

 leaf userDefinedNetworkType {

 type string;

 description "Textual information indicating network type, e.g. 'UTRAN'.";

 }

 }

 grouping SubNetworkGrp {

 uses Domain\_Grp;

 uses meas3gpp:SupportedPerfMetricGroupGrp;

 leaf-list setOfMcc {

 description "Set of Mobile Country Code (MCC).

 The MCC uniquely identifies the country of domicile

 of the mobile subscriber. MCC is part of the IMSI (3GPP TS 23.003)

 This list contains all the MCC values in subordinate object

 instances to this SubNetwork instance.

 See clause 2.3 of 3GPP TS 23.003 for MCC allocation principles.

 It shall be supported if there is more than one value in setOfMcc

 of the SubNetwork. Otherwise the support is optional.";

 type types3gpp:Mcc;

 }

 leaf priorityLabel {

 mandatory true;

 type uint32;

 }

 }

 list SubNetwork {

 key id;

 description "Represents a set of managed entities";

 uses top3gpp:Top\_Grp;

 container attributes {

 uses SubNetworkGrp;

 leaf-list parents {

 description "Reference to all containg SubNetwork instances

 in strict order from the root subnetwork down to the immediate

 parent subnetwork.

 If subnetworks form a containment hierarchy this is

 modeled using references between the child SubNetwork and the parent

 SubNetworks.

 This reference MUST NOT be present for the top level SubNetwork and

 MUST be present for other SubNetworks.";

 type leafref {

 path "../../../SubNetwork/id";

 }

 }

 leaf-list containedChildren{

 description "Reference to all directly contained SubNetwork instances.

 If subnetworks form a containment hierarchy this is

 modeled using references between the child SubNetwork and the parent

 SubNetwork.";

 type leafref {

 path "../../../SubNetwork/id";

 }

 }

 }

 uses meas3gpp:MeasurementSubtree {

 if-feature MeasurementsUnderSubNetwork;

 }

 uses subscr3gpp:SubscriptionControlSubtree {

 if-feature SubscriptionControlUnderSubNetwork;

 }

 uses fm3gpp:FmSubtree {

 if-feature FmUnderSubNetwork;

 }

 uses trace3gpp:TraceSubtree {

 if-feature TraceUnderSubNetwork;

 }

 yangmnt:mount-point children-of-SubNetwork {

 description "Mountpoint for ManagedElement";

 reference "RFC8528 YANG Schema Mount";

 }

 // augment external parts here

 }

}

<CODE ENDS>

***Next change***

## D.2.6a module \_3gpp-common-subscription-control.yang

<CODE BEGINS>

module \_3gpp-common-subscription-control {

 yang-version 1.1;

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

 prefix "subscr3gpp";

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

 organization "3GPP SA5";

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

 description "Defines IOCs for subscription and heartbeat control.";

 reference "3GPP TS 28.623

 Generic Network Resource Model (NRM)

 Integration Reference Point (IRP);

 Solution Set (SS) definitions

 3GPP TS 28.623";

 revision 2021-01-16 { reference "CR-0120"; }

 revision 2020-08-26 { reference "CR-0106"; }

 revision 2019-11-29 { reference "S5-197648 S5-197647 S5-197829 S5-197828"; }

 grouping NtfSubscriptionControlGrp {

 description "Attributes of a specific notification subscription";

 leaf notificationRecipientAddress {

 type string;

 mandatory true;

 }

 leaf-list notificationTypes {

 type string;

 description "Defines the types of notifications that are candidates

 for being forwarded to the notification recipient.

 If the notificationFilter attribute is not supported or not present

 all candidate notifications types are forwarded to the notification;

 discriminated by notificationFilter attribute.";

 }

 list scope {

 key "scopeType";

 min-elements 1;

 max-elements 1;

 description "Describes which object instances are selected with

 respect to a base object instance.";

 leaf scopeType {

 type enumeration {

 enum BASE\_ONLY;

 enum BASE\_ALL;

 enum BASE\_NTH\_LEVEL;

 enum BASE\_SUBTREE;

 }

 description "If the optional scopeLevel parameter is not supported

 or absent, allowed values of scopeType are BASE\_ONLY and BASE\_ALL.

 The value BASE\_ONLY indicates only the base object is selected.

 The value BASE\_ALL indicates the base object and all of its

 subordinate objects (incl. the leaf objects) are selected.

 If the scopeLevel parameter is supported and present, allowed

 values of scopeType are BASE\_ALL, BASE\_ONLY, BASE\_NTH\_LEVEL

 and BASE\_SUBTREE.

 The value BASE\_NTH\_LEVEL indicates all objects on the level,

 which is specified by the scopeLevel parameter, below the base

 object are selected. The base object is at scopeLevel zero.

 The value BASE\_SUBTREE indicates the base object and all of its

 subordinate objects down to and including the objects on the level,

 which is specified by the scopeLevel parameter, are selected.

 The base object is at scopeLevel zero.";

 }

 leaf scopeLevel {

 when '../scopeType = "BASE\_NTH\_LEVEL" or ../scopeType = "BASE\_SUBTREE"';

 type uint16;

 mandatory true;

 description "See description of scopeType.";

 }

 }

 leaf notificationFilter {

 type string;

 description "Defines a filter to be applied to candidate notifications

 identified by the notificationTypes attribute.

 If notificationFilter is present, only notifications that pass the

 filter criteria are forwarded to the notification recipient; all other

 notifications are discarded.

 The filter can be applied to any field of a notification.";

 }

 }

 grouping HeartbeatControlGrp {

 description "Attributes of HeartbeatControl. Note the triggerHeartbeatNtf

 attribute has no mapping in the present release.";

 leaf heartbeatNtfPeriod {

 type uint32;

 mandatory true;

 units seconds;

 description "Specifies the periodicity of heartbeat notification emission.

 The value of zero has the special meaning of stopping the heartbeat

 notification emission.";

 }

 }

 grouping SubscriptionControlSubtree {

 description "Contains notification subscription 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 NtfSubscriptionControl {

 description "A NtfSubscriptionControl instance represents the

 notification subscription of a particular notification recipient.

 The scope attribute is used to select managed object instances.

 The base object instance of the scope is the object instance

 name-containing the NtfSubscriptionControl instance.

 The notifications related to the selected managed object instances

 are candidates to be sent to the address specified by the

 notificationRecipientAddress attribute.

 The notificationType attribute and notificationFilter attribute

 allow MnS consumers to exercise control over which candidate

 notifications are sent to the notificationRecipientAddress.

 If the notificationType attribute is supported and present, its

 value identifies the

 types of notifications that are candidate to be sent to the

 notificationRecipientAddress. If the notificationType attribute is

 not supported or not present, all types of notifications are

 candidate to be sent to notificationRecipientAddress.

 If supported, the notificationFilter attribute defines a filter that

 is applied to the set of candidate notifications. Only candidate

 notifications that pass the filter criteria are sent to the

 notificationRecipientAddress. If the notificationFilter attribute is

 not supported all candidate notificatios are sent to the

 notificationRecipientAddress.

 To receive notifications, a MnS consumer has to create

 NtfSubscriptionControl object instancess on the MnS producer.

 A MnS consumer can create a subscription for another MnS consumer

 since it is not required the notificationRecipientAddress be his own

 address.

 When a MnS consumer does not wish to receive notifications any more

 the MnS consumer shall delete the corresponding NtfSubscriptionControl

 instance.

 Creation and deletion of NtfSubscriptionControl instances by MnS

 consumers is optional; when not supported, the NtfSubscriptionControl

 instances may be created and deleted by the system or be

 pre-installed.";

 key id;

 uses top3gpp:Top\_Grp;

 container attributes {

 uses NtfSubscriptionControlGrp;

 }

 list HeartbeatControl {

 description "MnS consumers (i.e. notification recipients) use heartbeat

 notifications to monitor the communication channels between them and

 data reporting MnS producers emitting notifications such as

 notifyNewAlarm and notifyFileReady.

 A HeartbeatControl instance allows controlling the emission of

 heartbeat notifications by MnS producers. The recipients of heartbeat

 notifications are specified by the notificationRecipientAddress

 attribute of the NtfSubscriptionControl instance containing the

 HeartbeatControl instance.

 Note that the MnS consumer managing the HeartbeatControl instance

 and the MnS consumer receiving the heartbeat notifications may not be

 the same.

 As a pre-condition for the emission of heartbeat notifications, a

 HeartbeatControl instance needs to be created. Creation of an instance

 with an initial non-zero value of the heartbeatNtfPeriod attribute

 triggers an immediate heartbeat notification emission. Creation of an

 instance with an initial zero value of the heartbeatPeriod attribute

 does not trigger an emission of a heartbeat notification. Deletion of

 an instance does not trigger an emission of a heartbeat notification.

 Once the instance is created, heartbeat notifications are emitted with

 a periodicity defined by the value of the heartbeatNtfPeriod

 attribute. No heartbeat notifications are emitted if the value is

 equal to zero. Setting a zero value to a non zero value, or a non zero

 value to a different non zero value, triggers an immediate heartbeat

 notification, that is the base for the new heartbeat period. Setting a

 non zero value to a zero value stops emitting heartbeats immediately;

 no final heartbeat notification is sent.

 Creation and deletion of HeartbeatControl instances by MnS Consumers

 is optional; when not supported, the HeartbeatControl instances may be

 created and deleted by the system or be pre-installed.

 Whether and when to emit heartbeat notifications is controlled by

 HeartbeatControl. Subscription for heartbeat is not supported via the

 NtfSubscriptionControl.";

 max-elements 1;

 key id;

 uses top3gpp:Top\_Grp;

 container attributes {

 uses HeartbeatControlGrp;

 }

 }

 }

 }

}

<CODE ENDS>

***Next change***

## D.2.10 module \_3gpp-common-trace.yang

<CODE BEGINS>

module \_3gpp-common-trace {

 yang-version 1.1;

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

 prefix "trace3gpp";

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

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

 import ietf-inet-types { prefix inet; }

 organization "3GPP SA5";

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

 description "Trace handling";

 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 2021-01-25 { reference "CR-0122"; }

 revision 2020-11-16 { reference "CR-0117"; }

 revision 2020-08-06 { reference "CR-0102"; }

 grouping TraceJobGrp {

 leaf tjJobType {

 type enumeration {

 enum IMMEDIATE\_MDT\_ONLY;

 enum LOGGED\_MDT\_ONLY;

 enum TRACE\_ONLY;

 enum IMMEDIATE\_MDT\_AND\_TRACE;

 enum RLF\_REPORT\_ONLY;

 enum RCEF\_REPORT\_ONLY;

 enum LOGGED\_MBSFN\_MDT;

 }

 default TRACE\_ONLY;

 description "Specifies the MDT mode and it specifies also whether the

 TraceJob represents only MDT, Logged MBSFN MDT, Trace or a combined

 Trace and MDT job. The attribute is applicable for Trace, MDT, RCEF and

 RLF reporting.";

 reference "Clause 5.9a of 3GPP TS 32.422 for additional details on the

 allowed values.";

 }

 list tjListOfInterfaces {

 key idx;

 must 'count(MSCServerInterfaces)+count(MGWInterfaces)+count(RNCInterfaces)'

 +'+count(SGSNInterfaces)+count(GGSNInterfaces)+count(S-CSCFInterfaces)'

 +'+count(P-CSCFInterfaces)+count(I-CSCFInterfaces)+count(MRFCInterfaces)'

 +'+count(MGCFInterfaces)+count(IBCFInterfaces)+count(E-CSCFInterfaces)'

 +'+count(BGCFInterfaces)+count(ASInterfaces)+count(HSSInterfaces)'

 +'+count(EIRInterfaces)+count(BM-SCInterfaces)+count(MMEInterfaces)'

 +'+count(SGWInterfaces)+count(PDN\_GWInterfaces)+count(eNBInterfaces)'

 +'+count(en-gNBInterfaces)+count(AMFInterfaces)+count(AUSFInterfaces)'

 +'+count(NEFInterfaces)+count(NRFInterfaces)+count(NSSFInterfaces)'

 +'+count(PCFInterfaces)+count(SMFInterfaces)+count(SMSFInterfaces)'

 +'+count(UDMInterfaces)+count(UPFInterfaces)+count(ng-eNBInterfaces)'

 +'+count(gNB-CU-CPInterfaces)+count(gNB-CU-UPInterfaces)+count(gNB-DUInterfaces)';

 description "Specifies the interfaces that need to be traced in the given

 ManagedEntityFunction.The attribute is applicable only for Trace. In

 case this attribute is not used, it carries a null semantic.";

 reference "Clause 5.5 of 3GPP TS 32.422 for additional details on the

 allowed values.";

 leaf idx { type uint32 ; }

 leaf-list MSCServerInterfaces {

 type enumeration {

 enum A ;

 enum Iu-CS ;

 enum Mc ;

 enum MAP-G ;

 enum MAP-B ;

 enum MAP-E ;

 enum MAP-F ;

 enum MAP-D ;

 enum MAP-C ;

 enum CAP ;

 }

 }

 leaf-list MGWInterfaces {

 type enumeration {

 enum Mc ;

 enum Nb-UP ;

 enum Iu-UP ;

 }

 }

 leaf-list RNCInterfaces {

 type enumeration {

 enum Iu-CS ;

 enum Iu-PS ;

 enum Iur ;

 enum Iub ;

 enum Uu ;

 }

 }

 leaf-list SGSNInterfaces {

 type enumeration {

 enum Gb ;

 enum Iu-PS ;

 enum Gn ;

 enum MAP-Gr ;

 enum MAP-Gd ;

 enum MAP-Gf ;

 enum Ge ;

 enum Gs ;

 enum S6d ;

 enum S4 ;

 enum S3 ;

 enum S13 ;

 }

 }

 leaf-list GGSNInterfaces {

 type enumeration {

 enum Gn ;

 enum Gi ;

 enum Gmb ;

 }

 }

 leaf-list S-CSCFInterfaces {

 type enumeration {

 enum Mw ;

 enum Mg ;

 enum Mr ;

 enum Mi ;

 }

 }

 leaf-list P-CSCFInterfaces {

 type enumeration {

 enum Gm ;

 enum Mw ;

 }

 }

 leaf-list I-CSCFInterfaces {

 type enumeration {

 enum Cx ;

 enum Dx ;

 enum Mg ;

 enum Mw ;

 }

 }

 leaf-list MRFCInterfaces {

 type enumeration {

 enum Mp ;

 enum Mr ;

 }

 }

 leaf-list MGCFInterfaces {

 type enumeration {

 enum Mg ;

 enum Mj ;

 enum Mn ;

 }

 }

 leaf-list IBCFInterfaces {

 type enumeration {

 enum Ix ;

 enum Mx ;

 }

 }

 leaf-list E-CSCFInterfaces {

 type enumeration {

 enum Mw ;

 enum Ml ;

 enum Mm ;

 enum Mi-Mg ;

 }

 }

 leaf-list BGCFInterfaces {

 type enumeration {

 enum Mi ;

 enum Mj ;

 enum Mk ;

 }

 }

 leaf-list ASInterfaces {

 type enumeration {

 enum Dh ;

 enum Sh ;

 enum ISC ;

 enum Ut ;

 }

 }

 leaf-list HSSInterfaces {

 type enumeration {

 enum MAP-C ;

 enum MAP-D ;

 enum Gc ;

 enum Gr ;

 enum Cx ;

 enum S6d ;

 enum S6a ;

 enum Sh ;

 }

 }

 leaf-list EIRInterfaces {

 type enumeration {

 enum MAP-F ;

 enum S13 ;

 enum MAP-Gf ;

 }

 }

 leaf-list BM-SCInterfaces {

 type enumeration {

 enum Gmb ;

 }

 }

 leaf-list MMEInterfaces {

 type enumeration {

 enum S1-MME ;

 enum S3 ;

 enum S6a ;

 enum S10 ;

 enum S11 ;

 enum S13 ;

 }

 }

 leaf-list SGWInterfaces {

 type enumeration {

 enum S4 ;

 enum S5 ;

 enum S8 ;

 enum S11 ;

 enum Gxc ;

 }

 }

 leaf-list PDN\_GWInterfaces {

 type enumeration {

 enum S2a ;

 enum S2b ;

 enum S2c ;

 enum S5 ;

 enum S6b ;

 enum Gx ;

 enum S8 ;

 enum SGi ;

 }

 }

 leaf-list eNBInterfaces {

 type enumeration {

 enum S1-MME ;

 enum X2 ;

 }

 }

 leaf-list en-gNBInterfaces {

 type enumeration {

 enum S1-MME ;

 enum X2 ;

 enum Uu ;

 enum F1-C ;

 enum E1 ;

 }

 }

 leaf-list AMFInterfaces {

 type enumeration {

 enum N1 ;

 enum N2 ;

 enum N8 ;

 enum N11 ;

 enum N12 ;

 enum N14 ;

 enum N15 ;

 enum N20 ;

 enum N22 ;

 enum N26 ;

 }

 }

 leaf-list AUSFInterfaces {

 type enumeration {

 enum N12 ;

 enum N13 ;

 }

 }

 leaf-list NEFInterfaces {

 type enumeration {

 enum N29 ;

 enum N30 ;

 enum N33 ;

 }

 }

 leaf-list NRFInterfaces {

 type enumeration {

 enum N27 ;

 }

 }

 leaf-list NSSFInterfaces {

 type enumeration {

 enum N22 ;

 enum N31 ;

 }

 }

 leaf-list PCFInterfaces {

 type enumeration {

 enum N5 ;

 enum N7 ;

 enum N15 ;

 }

 }

 leaf-list SMFInterfaces {

 type enumeration {

 enum N4 ;

 enum N7 ;

 enum N10 ;

 enum N11 ;

 enum S5-C ;

 }

 }

 leaf-list SMSFInterfaces {

 type enumeration {

 enum N20 ;

 enum N21 ;

 }

 }

 leaf-list UDMInterfaces {

 type enumeration {

 enum N8 ;

 enum N10 ;

 enum N13 ;

 enum N21 ;

 }

 }

 leaf-list UPFInterfaces {

 type enumeration {

 enum N4 ;

 }

 }

 leaf-list ng-eNBInterfaces {

 type enumeration {

 enum NG-C ;

 enum Xn-C ;

 enum Uu ;

 }

 }

 leaf-list gNB-CU-CPInterfaces {

 type enumeration {

 enum NG-C ;

 enum Xn-C ;

 enum Uu ;

 enum F1-C ;

 enum E1 ;

 enum X2-C ;

 }

 }

 leaf-list gNB-CU-UPInterfaces {

 type enumeration {

 enum E1 ;

 }

 }

 leaf-list gNB-DUInterfaces {

 type enumeration {

 enum F1-C ;

 }

 }

 }

 leaf-list tjListOfNeTypes {

 type enumeration {

 enum MSC\_SERVER;

 enum SGSN;

 enum MGW;

 enum GGSN;

 enum RNC;

 enum BM\_SC;

 enum MME;

 enum SGW;

 enum PGW;

 enum ENB;

 enum EN\_GNB;

 enum GNB\_CU\_CP;

 enum GNB\_CU\_UP;

 enum GNB\_DU;

 }

 description "Specifies in which type of ManagedFunction the trace should

 be activated. The attribute is applicable only for Trace with

 Signalling Based Trace activation. In case this attribute is not used,

 it carries a null semantic";

 reference "Clause 5.4 of 3GPP TS 32.422 for additional details on the

 allowed values";

 }

 leaf tjPLMNTarget {

 type string;

 mandatory true;

 description "Specifies which PLMN that the subscriber of the session to

 be recorded uses as selected PLMN. PLMN Target might differ from the

 PLMN specified in the Trace Reference";

 reference "Clause 5.9b of 3GPP TS 32.422";

 }

 leaf tjStreamingTraceConsumerURI {

 when './tjTraceReportingFormat = "STREAMING"';

 type inet:uri;

 mandatory true;

 description "URI of the Streaming Trace data reporting MnS consumer

 (a.k.a. streaming target).

 This attribute shall be present if file based trace data reporting is

 supported and tjTraceReportingFormat set to 'file based' or when

 tjJobType is set to Logged MDT or Logged MBSFN MDT.";

 reference "Clause 5.9 of 3GPP TS 32.422";

 }

 leaf tjTraceCollectionEntityAddress {

 when './tjTraceReportingFormat = "FILE\_BASED" or '

 +'./tjJobType = "LOGGED\_MDT\_ONLY" or ./tjJobType = "LOGGED\_MBSFN\_MDT"';

 type union {

 type inet:uri;

 type inet:ip-address;

 }

 mandatory true;

 description "Specifies the address of the Trace Collection Entity when

 the attribute tjTraceReportingFormat is configured for the file-based

 reporting. The attribute is applicable for both Trace and MDT.";

 reference "Clause 5.9 of 3GPP TS 32.422";

 }

 leaf tjTraceDepth {

 when './tjJobType = "TRACE\_ONLY" or ./tjJobType = "IMMEDIATE\_MDT\_AND\_TRACE"';

 type enumeration {

 enum MINIMUM;

 enum MEDIUM;

 enum MAXIMUM;

 enum VENDORMINIMUM;

 enum VENDORMEDIUM;

 enum VENDORMAXIMUM;

 }

 default MAXIMUM;

 description "Specifies how detailed information should be recorded in the

 Network Element. The Trace Depth is a paremeter for Trace Session level,

 i.e., the Trace Depth is the same for all of the NEs to be traced in

 the same Trace Session.

 The attribute is applicable only for Trace, otherwise it carries a null

 semantic.";

 reference "Clause 5.3 of 3GPP TS 32.422";

 }

 leaf tjTraceReference {

 type uint64;

 mandatory true;

 description "A globally unique identifier, which uniquely identifies the

 Trace Session that is created by the TraceJob.

 In case of shared network, it is the MCC and MNC of the Participating

 Operator that request the trace session that shall be provided.

 The attribute is applicable for both Trace and MDT.";

 }

 leaf tjTraceReportingFormat {

 type enumeration {

 enum FILE\_BASED;

 enum STREAMING;

 }

 default FILE\_BASED;

 description "Specifies the trace reporting format - streaming trace

 reporting or file-based trace reporting";

 reference "3GPP TS 32.422 clause 5.11";

 }

 list tjTraceTarget {

 key "targetIdType targetIdValue";

 max-elements 1;

 leaf targetIdType {

 type enumeration {

 enum IMSI;

 enum IMEI;

 enum IMEISV;

 enum PUBLIC\_ID;

 enum UTRAN\_CELL;

 enum E\_UTRAN\_CELL;

 enum NG\_RAN\_CELL;

 enum ENB;

 enum RNC;

 enum GNB;

 enum SUPI;

 }

 }

 leaf targetIdValue {

 type string;

 }

 description "Specifies the target object of the Trace and MDT. The

 attribute is applicable for both Trace and MDT. This attribute

 includes the ID type of the target as an enumeration and the ID value.

 The tjTraceTarget shall be public ID in case of a Management Based

 Activation is done to an ScscfFunction. The tjTraceTarget shall be

 cell only in case of the UTRAN cell traffic trace function.

 The tjTraceTarget shall be E-UtranCell only in case of E-UTRAN cell

 traffic trace function.The tjTraceTarget shall be either IMSI or

 IMEI(SV) if the Trace Session is activated to any of the following

 ManagedEntity(ies):

 - HssFunction

 - MscServerFunction

 - SgsnFunction

 - GgsnFunction

 - BmscFunction

 - RncFunction

 - MmeFunction

 The tjTraceTarget shall be IMSI if the Trace Session is activated to a

 ManagedEntity playing a role of ServinGWFunction.

 In case of signaling based Trace/MDT, the tjTraceTarget attribute shall be

 able to carry (IMSI or IMEI(SV)or SUPI), the tjMDTAreaScope attribute shall be

 able to carry a list of (cell or E-UtranCell or NRCellDU or TA/LA/RA).

 In case of management based Immediate MDT, the tjTraceTarget attribute

 shall be null value, the tjMDTAreaScope attribute shall carry a list of

 (Utrancell or E-UtranCell or NRCellDU).

 In case of management based Logged MDT, the tjTraceTarget attribute

 shall carry an eBs or a RNC or gNBs. The Logged MDT should be initiated on

 the specified eNB or RNC or gNB in tjTraceTarget. The tjMDTAreaScope attribute

 shall carry a list of (Utrancell or E-UtranCell or NRCellDU or TA/LA/RA).

 In case of RLF reporting, or RCEF reporting, the tjTraceTarget attribute

 shall be null value, the tjMDTAreaScope attribute shall carry one or

 list of eNBs/gNBs";

 reference "3GPP TS 32.422";

 }

 leaf tjTriggeringEvent {

 when './tjJobType = "TRACE" or ./tjJobType = "IMMEDIATE\_MDT\_AND\_TRACE"';

 type string ;

 mandatory true;

 description "Specifies the triggering event parameter of the trace session.

 The attribute is applicable only for Trace. In case this attribute is

 not used, it carries a null semantic.";

 reference "Clause 5.1 of 3GPP TS 32.422";

 }

 leaf tjMDTAnonymizationOfData {

 when ./tjMDTAreaScope ;

 type enumeration {

 enum NO\_IDENTITY;

 enum TAC\_OF\_IMEI;

 }

 default NO\_IDENTITY;

 description "Specifies level of MDT anonymization.";

 reference "3GPP TS 32.422 clause 5.10.12.";

 }

 list tjMDTAreaConfigurationForNeighCell {

 when './tjJobType = "LOGGED\_MDT\_ONLY"';

 key "idx";

 min-elements 1;

 leaf idx { type uint32 ; }

 description "It specifies the area for which UE is requested to perform

 measurement logging for neighbour cells which have list of frequencies.

 If it is not configured, the UE shall perform measurement logging for

 all the neighbour cells.

 Applicable only to NR Logged MDT.";

 reference "3GPP TS 32.422 clause 5.10.26.";

 leaf frequency {

 type string;

 }

 leaf cell {

 type string;

 }

 }

 leaf-list tjMDTAreaScope {

 type string;

 description "specifies MDT area scope when activates an MDT job.

 For RLF and RCEF reporting it specifies the eNB or list of eNBs where the

 RLF or RCEF reports should be collected.

 List of cells/TA/LA/RA for signaling based MDT or management based Logged

 MDT.

 List of cells for management based Immediate MDT.

 Cell, TA, LA, RA are mutually exclusive.

 One or list of eNBs for RLF and RCEFreporting";

 reference "Clause 5.10.2 of 3GPP TS 32.422";

 }

 leaf tjMDTCollectionPeriodRrmLte {

 when './tjJobType = "IMMEDIATE\_MDT\_ONLY" or ./tjJobType = "IMMEDIATE\_MDT\_AND\_TRACE"';

 type uint32 {

 range "250|500|1000|2000|3000|4000|6000|8000|12000|16000|20000|"

 +"24000|28000|32000|64000";

 }

 units milliseconds;

 description "Specifies the collection period for collecting RRM configured

 measurement samples for M2, M3 in LTE. The attribute is applicable only

 for Immediate MDT. In case this attribute is not used, it carries a

 null semantic.";

 reference "Clause 5.10.20 of 3GPP TS 32.422";

 }

 leaf tjMDTCollectionPeriodRrmUmts {

 when './tjJobType = "IMMEDIATE\_MDT\_ONLY" or ./tjJobType = "IMMEDIATE\_MDT\_AND\_TRACE"';

 type uint32 {

 range "1024|1280|2048|2560|5120|"

 +"10240|60000";

 }

 units milliseconds;

 description "Specifies the collection period for collecting RRM configured

 measurement samples for M3, M4, M5 in UMTS. The attribute is applicable

 only for Immediate MDT. In case this attribute is not used, it carries

 a null semantic";

 reference "Clause 5.10.21 of 3GPP TS 32.422";

 }

 leaf tjMDTCollectionPeriodRrmNR {

 when './tjJobType = "IMMEDIATE\_MDT\_ONLY" or ./tjJobType = "IMMEDIATE\_MDT\_AND\_TRACE"';

 type uint32 {

 range "1024|2048|5120|10240|60000";

 }

 units milliseconds;

 description "Specifies the collection period for collecting RRM configured

 measurement samples for M4, M5 in NR. The attribute is applicable only

 for Immediate MDT. In case this attribute is not used, it carries a

 null semantic.";

 reference "Clause 5.10.30 of 3GPP TS 32.422";

 }

 leaf tjMDTEventListForTriggeredMeasurement {

 when './tjJobType = "LOGGED\_MDT\_ONLY"';

 type enumeration {

 enum OUT\_OF\_COVERAGE ;

 enum A2\_EVENT ;

 }

 mandatory true;

 description "Specifies event types for event triggered measurement in the

 case of logged NR MDT. Each trace session may configure at most one

 event. The UE shall perform logging of measurements only upon certain

 condition being fulfilled:

 - Out of coverage.

 - A2 event.";

 reference "Clause 5.10.28 of 3GPP TS 32.422";

 }

 leaf tjMDTEventThreshold {

 type int64;

 description "Specifies the threshold which should trigger the reporting

 in case A2 event reporting in LTE or 1F/1l event in UMTS. The attribute

 is applicable only for Immediate MDT and when reportingTrigger is

 configured for A2 event in LTE or 1F event or 1l event in UMTS. In

 case this attribute is not used, it carries a null semantic.";

 reference "Clauses 5.10.7 and 5.10.7a of 3GPP TS 32.422";

 }

 leaf tjMDTListOfMeasurements {

 when './tjJobType = "IMMEDIATE\_MDT"';

 type int64;

 mandatory true;

 description "It specifies the UE measurements that shall be collected in

 an Immediate MDT job. The attribute is applicable only for Immediate MDT.

 In case this attribute is not used, it carries a null semantic.";

 reference "3GPP TS 32.422 clause 5.10.3";

 }

 leaf tjMDTLoggingDuration {

 when './tjJobType = "LOGGED\_MDT\_ONLY" or ./tjJobType = "LOGGED\_MBSFN\_MDT"';

 type uint32 {

 range "600|1200|2400|3600|5400|7200";

 }

 units seconds;

 mandatory true;

 description "Specifies how long the MDT configuration is valid at the

 UE in case of Logged MDT. The attribute is applicable only for

 Logged MDT and Logged MBSFN MDT. In case this attribute is not used, it

 carries a null semantic.";

 reference "5.10.9 of 3GPP TS 32.422";

 }

 leaf tjMDTLoggingInterval {

 when './tjJobType = "LOGGED\_MDT\_ONLY" or ./tjJobType = "LOGGED\_MBSFN\_MDT"';

 type uint32 {

 range "1280|2560|5120|10240|20480|"

 +"30720|40960|61440";

 }

 units milliseconds;

 mandatory true;

 description "Specifies the periodicty for Logged MDT. The attribute is

 applicable only for Logged MDT and Logged MBSFN MDT. In case this

 attribute is not used, it carries a null semantic";

 reference "5.10.8 of 3GPP TS 32.422";

 }

 leaf-list tjMDTMBSFNAreaList {

 when './tjJobType = "LOGGED\_MBSFN\_MDT"';

 type string;

 min-elements 1;

 max-elements 8;

 description "The MBSFN Area consists of a MBSFN Area ID and Carrier

 Frequency (EARFCN). The target MBSFN area List can have up to 8 entries.

 This parameter is applicable only if the job type is Logged MBSFN MDT.";

 reference "5.10.25 of 3GPP TS 32.422";

 }

 leaf tjMDTMeasurementPeriodLTE {

 when './tjJobType = "IMMEDIATE\_MDT\_ONLY" or ./tjJobType = "IMMEDIATE\_MDT\_AND\_TRACE"';

 type uint32 {

 range "1024|1280|2048|2560|5120|"

 +"10240|60000";

 }

 units milliseconds;

 mandatory true;

 description "It specifies the measurement period for the Data Volume and

 Scheduled IP throughput measurements for MDT taken by the eNB.

 The attribute is applicable only for Immediate MDT. In case this

 attribute is not used, it carries a null semantic.";

 reference "Clause 5.10.23 of 3GPP TS 32.422";

 }

 leaf tjMDTMeasurementPeriodUMTS {

 when './tjJobType = "IMMEDIATE\_MDT\_ONLY" or ./tjJobType = "IMMEDIATE\_MDT\_AND\_TRACE"';

 type uint32 {

 range "250|500|1000|2000|3000|4000|6000|8000|12000|16000|20000|"

 +"24000|28000|32000|64000";

 }

 units milliseconds;

 mandatory true;

 description "It specifies the measurement period for the Data Volume and

 Throughput measurements for MDT taken by RNC.

 The attribute is applicable only for Immediate MDT. In case this

 attribute is not used, it carries a null semantic.";

 reference "Clause 5.10.22 of 3GPP TS 32.422";

 }

 leaf tjMDTMeasurementQuantity {

 when './tjJobType = "IMMEDIATE\_MDT\_ONLY" or ./tjJobType = "IMMEDIATE\_MDT\_AND\_TRACE"';

 type uint64 ;

 mandatory true;

 description "It specifies the measurements that are collected in an MDT

 job for a UMTS MDT configured for event triggered reporting.";

 reference "Clause 5.10.15 of 3GPP TS 32.422";

 }

 list tjMDTPLMList {

 when './tjJobType = "LOGGED\_MDT\_ONLY"';

 key "mcc mnc";

 uses types3gpp:PLMNId;

 min-elements 1;

 max-elements 16;

 description "It indicates the PLMNs where measurement collection, status

 indication and log reporting is allowed.";

 reference "Clause 5.10.24 of 3GPP TS 32.422";

 }

 leaf tjMDTPositioningMethod {

 when './tjJobType = "IMMEDIATE\_MDT\_ONLY" or ./tjJobType = "IMMEDIATE\_MDT\_AND\_TRACE"';

 type enumeration {

 enum GNSS;

 enum E\_CELL\_ID;

 }

 mandatory true;

 description "It specifies what positioning method should be used in the

 MDT job.";

 reference "Clause 5.10.19 of 3GPP TS 32.422";

 }

 leaf tjMDTReportAmount {

 when './tjJobType = "IMMEDIATE\_MDT\_ONLY" and ./tjMDTReportingTrigger = "PERIODICAL"';

 type union {

 type uint32 {

 range "1|4|8|16|32|64" ;

 }

 type enumeration {

 enum INFINITY;

 }

 }

 mandatory true;

 description "It specifies the number of measurement reports that shall be

 taken for periodic reporting while the UE is in connected.

 The attribute is applicable only for Immediate MDT and when

 tjMDTReportingTrigger is configured for periodical measurements. In

 case this attribute is not used, it carries a null semantic.";

 reference "Clause 5.10.6 of 3GPP TS 32.422";

 }

 leaf tjMDTReportingTrigger {

 when './tjJobType = "IMMEDIATE\_MDT\_ONLY"';

 type enumeration {

 enum PERIODICAL;

 enum A2\_FOR\_LTE;

 enum 1F\_FOR\_UMTS;

 enum 1I\_FOR\_UMTS\_MCPS\_TDD;

 enum A2\_TRIGGERED\_PERIODIC\_FOR\_LTE;

 enum ALL\_CONFIGURED\_RRM\_FOR\_LTE;

 enum ALL\_CONFIGURED\_RRM\_FOR\_UMTS;

 }

 description "It specifies whether periodic or event based measurements

 should be collected.

 The attribute is applicable only for Immediate MDT and when the

 tjMDTListOfMeasurements is configured for M1 (for both UMTS and LTE)

 or M2 (only for UMTS). In case this attribute is not used, it carries

 a null semantic.";

 reference "Clause 5.10.4 of 3GPP TS 32.422";

 }

 leaf tjMDTReportInterval {

 when './tjJobType = "IMMEDIATE\_MDT\_ONLY" and ./tjMDTReportingTrigger = "PERIODICAL"';

 type uint32 {

 range "120|240|250|480|500|640|1000|1024|2000|2048|3000|4000|"

 +"5120|6000|8000|10240|12000|16000|20000|"

 +"24000|28000|32000|60000|64000|"

 +"360000|720000|1800000|3600000";

 }

 units milliseconds;

 mandatory true;

 description "It specifies the interval between the periodical measurements

 that shall be taken when the UE is in connected mode.

 The attribute is applicable only for Immediate MDT and when

 tjMDTReportingTrigger is configured for periodical measurements. In case

 this attribute is not used, it carries a null semantic.";

 reference "5.10.5 of 3GPP TS 32.422";

 }

 leaf tjMDTReportType {

 when './tjJobType = "LOGGED\_MDT\_ONLY"';

 type enumeration {

 enum PERIODICAL;

 enum EVENT\_TRIGGERED;

 }

 mandatory true;

 description "It specifies report type for logged NR MDT";

 reference "Clause 5.10.27 of 3GPP TS 32.422";

 }

 leaf tjMDTSensorInformation {

 type bits {

 bit BAROMETRIC\_PRESSURE;

 bit UE\_SPEED;

 bit UE\_ORIENTATION;

 }

 default "";

 description "It specifies which sensor information shall be included in

 logged NR MDT and immediate NR MDT measurement if they are available.

 The following sensor measurement can be included or excluded for the UE.";

 reference "Clause 5.10.29 of 3GPP TS 32.422";

 }

 leaf tjMDTTraceCollectionEntityID {

 when './tjJobType = "LOGGED\_MDT\_ONLY" or ./tjJobType = "LOGGED\_MBSFN\_MDT"';

 type uint8;

 mandatory true;

 description "It specifies the TCE Id which is sent to the UE in Logged MDT.";

 reference "Clause 5.10.11 of 3GPP TS 32.422";

 }

 }

 grouping TraceSubtree {

 description "Contains classes that manage Tracing.

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

 - SubNnetwork

 - ManagedElement

 - ManagedFunction

 If a YANG module wants to augment these classes/list/groupings they must

 augment all user classes!";

 list TraceJob {

 description "Represents the Trace Control and Configuration parameters of a

 particular Trace Job (see TS 32.421 and TS 32.422 for details).

 To activate Trace Jobs, a MnS consumer has to create TraceJob object

 instances on the MnS producer. A MnS consumer can activate a Trace Job

 for another MnS consumer since it is not required the value of

 tjTraceCollectionEntityAddress or tjStreamingTraceConsumerUri to be his

 own.

 When a MnS consumer wishes to deactivate a Trace Job, the MnS consumer

 shall delete the corresponding TraceJob instance.

 For details of management Trace Job activation/deactivation see clause

 4.1.1.1.2 of TS 32.422.

 Creation and deletion of TraceJob instances by MnS consumers is optional;

 when not supported, the TraceJob instances may be created and deleted by

 the system or be pre-installed.";

 key id;

 uses top3gpp:Top\_Grp ;

 container attributes {

 uses TraceJobGrp ;

 }

 }

 }

}

<CODE ENDS>

***End of changes***