**3GPP TSG- Meeting #**

**, , -**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| *CR-Form-v12.1* | | | | | | | | |
| **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 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-15 (Release 15) Rel-16 (Release 16) Rel-17 (Release 17) Rel-18 (Release 18)* | |
|  |  | | | | | | | | | |
| ***Reason for change:*** | | The definition of S-NSSAI in the YANG SS in incorrect   * S-SNSSAI is comprised of SST and optional SD, but current definition in YANG SS uses YANG union construct. * Stage 2 for S-NSSAI is defined in 28.541 (5G NRM) rather than 28.622, so the correct location for this definition is in TS 28.541. | | | | | | | | |
|  | |  | | | | | | | | |
| ***Summary of change:*** | | Removal of S-NSSAI related definitions from the Generic NRM YANG SS  Corerction of a YANG syntax error in \_3gpp-common-measurements. (previously not detected by pyang) | | | | | | | | |
|  | |  | | | | | | | | |
| ***Consequences if not approved:*** | | * Implementations would use incorrect S-NSSAI definition. * YANG definition would be difficult to find, since stage 2 is in 28.541. * Compilation error in \_3gpp-common-measurements | | | | | | | | |
|  | |  | | | | | | | | |
| ***Clauses affected:*** | | D.2.4, D.2.8 | | | | | | | | |
|  | |  | | | | | | | | |
|  | | **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 28.541 CR 0411 (S5-206036) | | |
|  | |  | | | | | | | | |
| ***Other comments:*** | | Forge link: TBD  See also above related CR to add corrected S-NSSAI definition for YANG SS in TS 28.541 and also update related YANG where the S-NSSAI type is used.  Forge link: <https://forge.3gpp.org/rep/sa5/MnS/tree/S5-206036_Rel-16_CR_28.541_YANG_improvements>  The link is provided towards another CR, as only together with those changes can this CR be truly tested. | | | | | | | | |
|  | |  | | | | | | | | |
| ***This CR's revision history:*** | |  | | | | | | | | |

***First change***

D.2.4 module \_3gpp-common-measurements.yang

module \_3gpp-common-measurements {

yang-version 1.1;

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

prefix "meas3gpp";

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

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

organization "3GPP SA5";

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

description "Defines Measurement and KPI related groupings

Any list/class intending to use this should include 2 or 3 uses statements

controlled by a feature:

A)

+++ feature MeasurementsUnderMyClass {

+++ description 'Indicates whether measurements and/or KPIs are supported

+++ for this class.';

+++ }

B) include the attribute measurementsList and/or kPIsList indicating the

supported measurment and KPI types and GPs. Note that for classes

inheriting from ManagedFunction, EP\_RP or SubNetwork these attributes are

already inherited, so there is no need to include them once more. E.g.

+++ grouping MyClassGrp {

+++ uses meas3gpp:SupportedPerfMetricGroup;

+++ }

C) include the class PerfmetricJob to control the measurements/KPIs. E.g.

list MyClass {

container attributes {

uses MyClassGrp;

}

+++ uses meas3gpp:MeasurementSubtree {

+++ if-feature MeasurementsUnderMyClass ;

+++ }

}

Measurements can be contained under ManagedElement, SubNetwork, or

any list representing a class inheriting from Subnetwork or

ManagedFunction. Note: KPIs will only be supported under SubNetwork";

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 2020-11-06 { reference "CR-0118"; }

revision 2020-09-04 { reference "CR-000107"; }

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

revision 2020-05-31 { reference "CR-0084"; }

revision 2020-03-11 { reference "S5-201581, SP-200229"; }

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

revision 2019-10-28 { reference "S5-193516"; }

revision 2019-06-17 { }

grouping ThresholdInfoGrp {

description "Defines a single threshold level.";

leaf-list measurementTypes {

type string;

description "The Measurement type can be those specified in TS 28.552,

TS 32.404 and can be those specified by other SDOs or can be

vendor-specific.";

}

leaf thresholdLevel {

type uint64;

mandatory true;

description "Number (key) for a single threshold in the threshold list

applicable to the monitored performance metric.";

}

leaf thresholdDirection {

type enumeration {

enum UP;

enum DOWN;

enum UP\_AND\_DOWN;

}

must '. = "UP\_AND\_DOWN" or not(../hysteresis)' {

error-message "In case a threshold with hysteresis is configured, the "

+"threshold direction attribute shall be set to 'UP\_AND\_DOWN'.";

}

mandatory true;

description "Direction of a threshold indicating the direction for which

a threshold crossing triggers a threshold.

When the threshold direction is configured to 'UP', the associated

treshold is triggered only when the performance metric value is going

up upon reaching or crossing the threshold value. The treshold is not

triggered, when the performance metric is going down upon reaching or

crossing the threshold value.

Vice versa, when the threshold direction is configured to 'DOWN', the

associated treshold is triggered only when the performance metric is

going down upon reaching or crossing the threshold value. The treshold

is not triggered, when the performance metric is going up upon reaching

or crossing the threshold value.

When the threshold direction is set to 'UP\_AND\_DOWN' the treshold is

active in both direcions.

In case a threshold with hysteresis is configured, the threshold

direction attribute shall be set to 'UP\_AND\_DOWN'.";

}

leaf thresholdValue {

type union {

type int64;

type decimal64 {

fraction-digits 2;

}

}

mandatory true;

description "Value against which the monitored performance metric is

compared at a threshold level in case the hysteresis is zero";

}

leaf hysteresis {

type union {

type uint64;

type decimal64 {

fraction-digits 2;

range "0..max";

}

}

description "Hysteresis of a threshold. If this attribute is present

the monitored performance metric is not compared against the

threshold value as specified by the thresholdValue attribute but

against a high and low threshold value given by

threshold-high = thresholdValue + hysteresis

threshold-low = thresholdValue - hysteresis

When going up, the threshold is triggered when the performance metric

reaches or crosses the high threshold value. When going down, the

hreshold is triggered when the performance metric reaches or crosses

the low threshold value.

A hysteresis may be present only when the monitored performance

metric is not of type counter that can go up only. If present

for a performance metric of type counter, it shall be ignored.";

}

}

grouping SupportedPerfMetricGroupGrp {

list SupportedPerfMetricGroup {

config false;

description "Captures a group of supported performance metrics and

associated parameters related to their production and reporting.

A SupportedPerfMetricGroup attribute which is part of an MOI may

define performanceMetrics for any MOI under the subtree contained

under that MOI, e.g. SupportedPerfMetricGroup on a ManagedElement

can specify supported metrics for contained ManagedFunctions

like a GNBDUFunction.";

leaf-list performanceMetrics {

type string;

min-elements 1;

description "Performance metrics include measurements defined in

TS 28.552 and KPIs defined in TS 28.554. Performance metrics can

also be those specified by other SDOs or vendor specific metrics.

Performance metrics are identfied with their names. A name can also

identify a vendor specific group of performance metrics.

For measurements defined in TS 28.552 the name is constructed as

follows:

- 'family.measurementName.subcounter' for measurement types with

subcounters

- 'family.measurementName' for measurement types without subcounters

- 'family' for measurement families

For KPIs defined in TS 28.554 the name is defined in the KPI

definitions template as the component designated with e).";

}

leaf-list granularityPeriods {

type uint32 {

range 1..max ;

}

units seconds;

}

leaf-list reportingMethods {

type enumeration {

enum FILE\_BASED\_LOC\_SET\_BY\_PRODUCER;

enum FILE\_BASED\_LOC\_SET\_BY\_CONSUMER;

enum STREAM\_BASED;

}

min-elements 1;

}

leaf-list monitorGranularityPeriods {

type uint32 {

range 1..max ;

}

units seconds;

description "Granularity periods supported for the monitoring of

associated measurement types for thresholds";

}

}

}

grouping PerfMetricJobGrp {

description "Represents the attributtes of the IOC PerfMetricJob";

leaf administrativeState {

default UNLOCKED;

type types3gpp:AdministrativeState ;

description "Enable or disables production of the metrics";

}

leaf operationalState {

config false;

mandatory true;

type types3gpp:OperationalState ;

description "Indicates whether the PerfMetricJob is working.";

}

leaf perfMetricJobGroupId {

type string;

description "Identifies members of a PerfMetricJob group. For the

stream based reporting method this reference shall be present.";

}

leaf-list performanceMetrics {

type string;

min-elements 1;

description "Performance metrics include measurements defined in

TS 28.552 and KPIs defined in TS 28.554. Performance metrics can

also be those specified by other SDOs or vendor specific metrics.

Performance metrics are identfied with their names. A name can also

identify a vendor specific group of performance metrics.

For measurements defined in TS 28.552 the name is constructed as

follows:

- 'family.measurementName.subcounter' for measurement types with

subcounters

- 'family.measurementName' for measurement types without subcounters

- 'family' for measurement families

For KPIs defined in TS 28.554 the name is defined in the KPI

definitions template as the component designated with e).";

}

leaf granularityPeriod {

type uint32 {

range 1..max ;

}

units seconds;

mandatory true;

description "Granularity period used to produce measurements. The value

must be one of the supported granularity periods for the metric.

For measurements of type counter this is the period at which samples

of the internal counter value, that is incremented with every event

occurance, are taken.

For measurements of type gauge, this is period, over which the mean

value of the measured variable is calculated. The mean value is

then taken as sample.";

}

leaf-list objectInstances {

type types3gpp:DistinguishedName;

}

leaf-list rootObjectInstances {

type types3gpp:DistinguishedName;

description "Each object instance designates the root of a subtree that

contains the root object and all descendant objects.";

}

choice reportingCtrl {

mandatory true;

description "This choice defines the method for reporting collected

performance metrics to MnS consumers as well as the parameters for

configuring the reporting function. It is a choice between the control

parameter required for the reporting methods, whose presence selects

the reporting method as follows:

- When only the fileReportingPeriod attribute is present, the MnS

producer shall store files on the MnS producer at a location selected

by the MnS producer and inform the MnS consumer about the availability

of new files and the file location using the notifyFileReady

notification.

- When only the fileReportingPeriod and fileLocation attributes are

present, the MnS producer shall store the files on the MnS consumer at

the location specified by fileLocation. No notification is emitted by

the MnS producer.

- When only the streamTarget attribute is present, the MnS producer

shall stream the data to the location specified by streamTarget.

For the file-based reporting methods the fileReportingPeriod attribute

specifies the time window during which collected measurements are stored

into the same file before the file is closed and a new file is opened.";

case file-based-reporting {

leaf fileReportingPeriod {

type uint32 {

range 1..max;

}

units minutes;

must '(number(.)\*"60") mod number(../granularityPeriod) = "0"' {

error-message

"The time-period must be a multiple of the granularityPeriod.";

}

mandatory true;

description "For the file-based reporting method this is the time

window during which collected measurements are stored into the same

file before the file is closed and a new file is opened.

The time-period must be a multiple of the granularityPeriod.

Applicable when the file-based reporting method is supported";

}

leaf fileLocation {

type string ;

description "Applicable and must be present when the file-based

reporting method is supported, and the files are stored on the MnS

consumer.";

}

}

case stream-based-reporting {

leaf streamTarget {

type string;

mandatory true;

description "Applicable when stream-based reporting method is

supported.";

}

}

}

}

grouping ThresholdMonitorGrp {

description "A threshold monitor that is created by the consumer for

the monitored entities whose measurements are required by consumer

to monitor.";

leaf administrativeState {

default UNLOCKED;

type types3gpp:AdministrativeState ;

description "Enables or disables the ThresholdMonitor.";

}

leaf operationalState {

config false;

mandatory true;

type types3gpp:OperationalState ;

description "Indicates whether the ThresholdMonitor is working.";

}

list thresholdInfoList {

key idx;

min-elements 1;

leaf idx { type uint32 ; }

uses ThresholdInfoGrp;

}

leaf monitorGranularityPeriod {

type uint32 {

range "1..max";

}

units second;

mandatory true;

description " Granularity period used to monitor measurements for

threshold crossings. ";

}

leaf-list objectInstances {

type types3gpp:DistinguishedName;

}

leaf-list rootObjectInstances {

type types3gpp:DistinguishedName;

description "Each object instance designates the root of a subtree that

contains the root object and all descendant objects.";

}

}

grouping MeasurementSubtree {

description "Contains classes that define measurements.

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!

If a class uses this grouping in its list it shall also use the

grouping SupportedPerfMetricGroupGrp to add SupportedPerfMetricGroup as

an attribute to its grouping";

list PerfMetricJob {

description "This IOC represents a performance metric production job. It

can be name-contained by SubNetwork, ManagedElement, or ManagedFunction.

To activate the production of the specified performance metrics, a MnS

consumer needs to create a PerfMetricJob instance on the MnS producer

and ensure that the adminState is sUNLOCKED>.

For ultimate deactivation of metric production, the MnS consumer should

delete the job to free up resources on the MnS producer.

For temporary suspension of metric production, the MnS consumer can

manipulate the value of the administrative state attribute. The MnS

producer may disable metric production as well, for example in overload

situations. This situation is indicated by the MnS producer with setting

the operational state attribute to disabled. When production is resumed

the operational state is set again to enabled.

The perfMetricJobGroupId is a common reference across all members of a

PerfMetricJob group. A group contains related PerfMetricJob instances.

The attribute performanceMetrics defines the performance metrics to be

produced and the attribute granularityPeriod defines the granularity

period to be applied.

All object instances below and including the instance name-containing

the PerfMetricJob (base object instance) are scoped for performance

metric production. Performance metrics are produced only on those object

instances whose object class matches the object class associated to the

performance metrics to be produced.

The attributes objectInstances and rootObjectInstances allow to restrict

the scope. When the attribute objectInstances is present, only the object

instances identified by this attribute are scoped. When the attribute

rootObjectInstances is present, then the subtrees whose root objects are

identified by this attribute are scoped. Both attributes may be present

at the same time meaning the total scope is equal to the sum of both

scopes. Object instances may be scoped by both the objectInstances and

rootObjectInstances attributes. This shall not be considered as an error

by the MnS producer.

When the performance metric requires performance metric production on

multiple managed objects, which is for example the case for KPIs, the

MnS consumer needs to ensure all required objects are scoped. Otherwise

a PerfMetricJob creation request shall fail.

The attribute reportingCtrl specifies the method and associated control

parameters for reporting the produced measurements to MnS consumers.

Three methods are available: file-based reporting with selection of the

file location by the MnS producer, file-based reporting with selection

of the file location by the MnS consumer and stream-based reporting.

A PerfMetricJob creation request shall fail, when the requested

performance metrics, the requested granularity period, the requested

repoting method, or the requested combination thereof is not supported

by the MnS producer.

Creation and deletion of PerfMetricJob instances by MnS consumers is

optional; when not supported, PerfMetricJob instances may be created and

deleted by the system or be pre-installed.";

key id;

uses top3gpp:Top\_Grp ;

container attributes {

uses PerfMetricJobGrp ;

}

}

list ThresholdMonitor {

key id;

description "Represents a threshold monitor for performance metrics.

It can be contained by SubNetwork, ManagedElement, or ManagedFunction.

A threshold monitor checks for threshold crossings of performance metric

values and generates a notification when that happens.

To activate threshold monitoring, a MnS consumer needs to create a

ThresholdMonitor instance on the MnS producer. For ultimate deactivation

of threshold monitoring, the MnS consumer should delete the monitor to

free up resources on the MnS producer.

For temporary suspension of threshold monitoring, the MnS consumer can

manipulate the value of the administrative state attribute. The MnS

producer may disable threshold monitoring as well, for example in

overload situations. This situation is indicated by the MnS producer with

setting the operational state attribute to disabled. When monitoring is

resumed the operational state is set again to enabled.

All object instances below and including the instance containing the

ThresholdMonitor (base object instance) are scoped for performance

metric production. Performance metrics are monitored only on those

object instances whose object class matches the object class associated

to the performance metrics to be monitored.

The optional attributes objectInstances and rootObjectInstances allow to

restrict the scope. When the attribute objectInstances is present, only

the object instances identified by this attribute are scoped. When the

attribute rootObjectInstances is present, then the subtrees whose root

objects are identified by this attribute are scoped. Both attributes may

be present at the same time meaning the total scope is equal to the sum

of both scopes. Object instances may be scoped by both the objectInstances

and rootObjectInstances attributes. This shall not be considered as an

error by the MnS producer.

Multiple thresholds can be defined for multiple performance metric sets

in a single monitor using thresholdInfoList. The attribute

monitorGranularityPeriod defines the granularity period to be applied.

Each threshold is identified with a number (key) called thresholdLevel.

A threshold is defined using the attributes thresholdValue ,

thresholdDirection and hysteresis.

When hysteresis is absent or carries no information, a threshold is

triggered when the thresholdValue is reached or crossed. When hysteresis

is present, two threshold values are specified for the threshold as

follows: A high treshold value equal to the threshold value plus the

hysteresis value, and a low threshold value equal to the threshold value

minus the hysteresis value. When the monitored performance metric

increases, the threshold is triggered when the high threshold value is

reached or crossed. When the monitored performance metric decreases, the

threshold is triggered when the low threshold value is reached or crossed.

The hsyteresis ensures that the performance metric value can oscillate

around a comparison value without triggering each time the threshold when

the threshold value is crossed.

Using the thresholdDirection attribute a threshold can be configured in

such a manner that it is triggered only when the monitored performance

metric is going up or down upon reaching or crossing the threshold.

A ThresholdMonitor creation request shall be rejected, if the performance

metrics requested to be monitored, the requested granularity period, or

the requested combination thereof is not supported by the MnS producer.

A creation request may fail, when the performance metrics requested to be

monitored are not produced by a PerfMetricJob.

Creation and deletion of ThresholdMonitor instances by MnS consumers is

optional; when not supported, ThresholdMonitor instances may be created

and deleted by the system or be pre-installed.";

uses top3gpp:Top\_Grp ;

container attributes {

uses ThresholdMonitorGrp ;

}

}

}

}

***First change***

## D.2.8 module \_3gpp-common-yang-types.yang

module \_3gpp-common-yang-types {

yang-version 1.1;

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

prefix "types3gpp";

import ietf-inet-types { prefix inet; }

import ietf-yang-types { prefix yang; }

organization "3GPP SA5";

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

description "The model defines a YANG mapping of the top level

information classes used for management of 5G networks and

network slicing.";

reference "3GPP TS 28.541";

revision 2020-11-06 {

description "Removed incorrect S-NSSAI definitions.";

reference "CR-0118";

}

revision 2020-03-10 {

description "Removed faulty when statements.";

reference “SP-200229”;

}

revision 2019-10-25 {

description "Added ManagedNFProfile.";

reference "S5-194457";

}

revision 2019-10-16 {

description "Added SAP and usageState.";

reference "S5-193518";

}

revision 2019-06-23 {

reference "Initial version.";

}

grouping ManagedNFProfile {

description "Defines profile for managed NF";

reference "3GPP TS 23.501";

leaf idx { type uint32 ; }

leaf nfInstanceID {

config false;

mandatory true;

type yang:uuid ;

description "This parameter defines profile for managed NF.

The format of the NF Instance ID shall be a

Universally Unique Identifier (UUID) version 4,

as described in IETF RFC 4122 " ;

}

leaf-list nfType {

config false;

min-elements 1;

type NfType;

description "Type of the Network Function" ;

}

leaf hostAddr {

mandatory true;

type inet:host ;

description "Host address of a NF";

}

leaf authzInfo {

type string ;

description "This parameter defines NF Specific Service authorization

information. It shall include the NF type (s) and NF realms/origins

allowed to consume NF Service(s) of NF Service Producer.";

reference "See TS 23.501" ;

}

leaf location {

type string ;

description "Information about the location of the NF instance

(e.g. geographic location, data center) defined by operator";

reference "TS 29.510" ;

}

leaf capacity {

mandatory true;

type uint16 ;

description "This parameter defines static capacity information

in the range of 0-65535, expressed as a weight relative to other

NF instances of the same type; if capacity is also present in the

nfServiceList parameters, those will have precedence over this value.";

reference "TS 29.510" ;

}

leaf nFSrvGroupId {

type string ;

description "This parameter defines identity of the group that is

served by the NF instance.

May be config false or true depending on the ManagedFunction.

Config=true for Udrinfo. Config=false for UdmInfo and AusfInfo.

Shall be present if ../nfType = UDM or AUSF or UDR. ";

reference "TS 29.510" ;

}

leaf-list supportedDataSetIds {

type enumeration {

enum SUBSCRIPTION;

enum POLICY;

enum EXPOSURE;

enum APPLICATION;

}

description "List of supported data sets in the UDR instance.

May be present if ../nfType = UDR";

reference "TS 29.510" ;

}

leaf-list smfServingAreas {

type string ;

description "Defines the SMF service area(s) the UPF can serve.

Shall be present if ../nfType = UPF";

reference "TS 29.510" ;

}

leaf priority {

type uint16;

description "This parameter defines Priority (relative to other NFs

of the same type) in the range of 0-65535, to be used for NF selection;

lower values indicate a higher priority. If priority is also present

in the nfServiceList parameters, those will have precedence over

this value. Shall be present if ../nfType = AMF ";

reference "TS 29.510" ;

}

}

typedef usageState {

type enumeration {

enum IDLE;

enum ACTIVE;

enum BUSY;

}

description "It describes whether or not the resource is actively in

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

capacity for additional users at that instant. The value is READ-ONLY.";

reference "ITU T Recommendation X.731";

}

grouping SAP {

leaf host {

type inet:host;

mandatory true;

}

leaf port {

type inet:port-number;

mandatory true;

}

description "Service access point.";

reference "TS 28.622";

}

typedef Mcc {

description "The mobile country code consists of three decimal digits,

The first digit of the mobile country code identifies the geographic

region (the digits 1 and 8 are not used):";

type string {

pattern '[02-79][0-9][0-9]';

}

reference "3GPP TS 23.003 subclause 2.2 and 12.1";

}

typedef Mnc {

description "The mobile network code consists of two or three

decimal digits (for example: MNC of 001 is not the same as MNC of 01)";

type string {

pattern '[0-9][0-9][0-9]|[0-9][0-9]';

}

reference "3GPP TS 23.003 subclause 2.2 and 12.1";

}

grouping PLMNId {

leaf mcc {

mandatory true;

type Mcc;

}

leaf mnc {

mandatory true;

type Mnc;

}

reference "TS 23.658";

}

typedef Nci {

description "NR Cell Identity. The NCI shall be of fixed length of 36 bits

and shall be coded using full hexadecimal representation.

The exact coding of the NCI is the responsibility of each PLMN operator";

reference "TS 23.003";

type union {

type string {

length 36;

pattern '[01]+';

}

type string {

length 9;

pattern '[a-fA-F0-9]\*';

}

}

}

typedef OperationalState {

reference "3GPP TS 28.625 and ITU-T X.731";

type enumeration {

enum DISABLED {

value 0;

description "The resource is totally inoperable.";

}

enum ENABLED {

value 1;

description "The resource is partially or fully operable.";

}

}

}

typedef AdministrativeState {

reference "3GPP TS 28.625 and ITU-T X.731";

type enumeration {

enum LOCKED {

value 0;

description "The resource is administratively prohibited from performing

services for its users.";

}

enum UNLOCKED {

value 1;

description "The resource is administratively permitted to perform

services for its users. This is independent of its inherent

operability.";

}

enum SHUTTINGDOWN {

value 2;

description "Use of the resource is administratively permitted to

existing instances of use only. While the system remains in

the shutting down state the manager or the managed element

may at any time cause the resource to transition to the

locked state.";

}

}

}

typedef AvailabilityStatus {

type enumeration {

enum IN\_TEST;

enum FAILED;

enum POWER\_OFF;

enum OFF\_LINE;

enum OFF\_DUTY;

enum DEPENDENCY;

enum DEGRADED;

enum NOT\_INSTALLED;

enum LOG\_FULL;

}

}

typedef CellState {

type enumeration {

enum IDLE;

enum INACTIVE;

enum ACTIVE;

}

}

typedef Nrpci {

type uint32;

description "Physical Cell Identity (PCI) of the NR cell.";

reference "TS 36.211 subclause 6.11";

}

typedef Tac {

type int32 {

range 0..16777215 ;

}

description "Tracking Area Code";

reference "TS 23.003 clause 19.4.2.3";

}

typedef AmfRegionId {

type union {

type uint8 ;

type string {

length 8;

pattern '[01]\*';

}

}

reference "clause 2.10.1 of 3GPP TS 23.003";

}

typedef AmfSetId {

type union {

type uint16 {

range '0..1023';

}

type string {

length 8;

pattern '[01]\*';

}

}

reference "clause 2.10.1 of 3GPP TS 23.003";

}

typedef AmfPointer {

type union {

type uint8 {

range '0..63';

}

type string {

length 6;

pattern '[01]\*';

}

}

reference "clause 2.10.1 of 3GPP TS 23.003";

}

grouping AmfIdentifier {

leaf amfRegionId {

type AmfRegionId;

}

leaf amfSetId {

type AmfSetId;

}

leaf amfPointer {

type AmfPointer;

}

description "The AMFI is constructed from an AMF Region ID,

an AMF Set ID and an AMF Pointer.

The AMF Region ID identifies the region,

the AMF Set ID uniquely identifies the AMF Set within the AMF Region, and

the AMF Pointer uniquely identifies the AMF within the AMF Set. ";

}

// type definitions especially for core NFs

typedef NfType {

type enumeration {

enum NRF;

enum UDM;

enum AMF;

enum SMF;

enum AUSF;

enum NEF;

enum PCF;

enum SMSF;

enum NSSF;

enum UDR;

enum LMF;

enum GMLC;

enum 5G\_EIR;

enum SEPP;

enum UPF;

enum N3IWF;

enum AF;

enum UDSF;

enum BSF;

enum CHF;

}

}

typedef NotificationType {

type enumeration {

enum N1\_MESSAGES;

enum N2\_INFORMATION;

enum LOCATION\_NOTIFICATION;

}

}

typedef Load {

description "Latest known load information of the NF, percentage ";

type uint8 {

range 0..100;

}

}

typedef N1MessageClass {

type enumeration {

enum 5GMM;

enum SM;

enum LPP;

enum SMS;

}

}

typedef N2InformationClass {

type enumeration {

enum SM;

enum NRPPA;

enum PWS;

enum PWS\_BCAL;

enum PWS\_RF;

}

}

grouping DefaultNotificationSubscription {

leaf notificationType {

type NotificationType;

}

leaf callbackUri {

type inet:uri;

}

leaf n1MessageClass {

type N1MessageClass;

}

leaf n2InformationClass {

type N2InformationClass;

}

}

grouping Ipv4AddressRange {

leaf start {

type inet:ipv4-address;

}

leaf end {

type inet:ipv4-address;

}

}

grouping Ipv6PrefixRange {

leaf start {

type inet:ipv6-prefix;

}

leaf end {

type inet:ipv6-prefix;

}

}

typedef NsiId {

type string;

}

typedef UeMobilityLevel {

type enumeration {

enum STATIONARY;

enum NOMADIC;

enum RESTRICTED\_MOBILITY;

enum FULLY\_MOBILITY;

}

}

typedef ResourceSharingLevel {

type enumeration {

enum SHARED;

enum NOT\_SHARED;

}

}

typedef TxDirection {

type enumeration {

enum DL;

enum UL;

enum DL\_AND\_UL;

}

}

grouping AddressWithVlan {

leaf ipAddress {

type inet:ip-address;

}

leaf vlanId {

type uint16;

}

}

typedef DistinguishedName { // TODO is this equivalent to TS 32.300 ?

type string {

pattern '([a-zA-Z][a-zA-Z0-9-]\*=(\\( |#|\\|>|<|;|"|\+|,|[a-fA-F0-9]{2})|[^\\><;"+,# ])'

+ '((\\( |#|\\|>|<|;|"|\+|,|[a-fA-F0-9]{2})|[^\\><;"+,])\*'

+ '(\\( |#|\\|>|<|;|"|\+|,|[a-fA-F0-9]{2})|[^\\><;"+, ]))?'

+ '[,\+])\*[a-zA-Z][a-zA-Z0-9-]\*=(\\( |#|\\|>|<|;|"|\+|,|[a-fA-F0-9]{2})|[^\\><;"+,# ])'

+ '((\\( |#|\\|>|<|;|"|\+|,|[a-fA-F0-9]{2})'

+ '|[^\\><;"+,])\*(\\( |#|\\|>|<|;|"|\+|,|[a-fA-F0-9]{2})|[^\\><;"+, ]))?';

}

description "Represents the international standard for the representation

of Distinguished Name (RFC 4512).

The format of the DistinguishedName REGEX is:

{AttributeType = AttributeValue}

AttributeType consists of alphanumeric and hyphen (OIDs not allowed).

All other characters are restricted.

The Attribute value cannot contain control characters or the

following characters : \\ > < ; \" + , (Comma) and White space

The Attribute value can contain the following characters if they

are excaped : \\ > < ; \" + , (Comma) and White space

The Attribute value can contain control characters if its an escaped

double digit hex number.

Examples could be

UID=nobody@example.com,DC=example,DC=com

CN=John Smith,OU=Sales,O=ACME Limited,L=Moab,ST=Utah,C=US";

reference "RFC 4512 Lightweight Directory Access Protocol (LDAP):

Directory Information Models";

} // recheck regexp it doesn't handle posix [:cntrl:]

typedef QOffsetRange {

type int8 {

range "-24 | -22 | -20 | -18 | -16 | -14 | -12 | -10 | -8 | -6 | " +

" -5 | -4 | -3 | -2 | -1 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 8 | 10 | " +

" 12 | 14 | 16 | 18 | 20 | 22 | 24";

}

units dB;

}

}

***End of changes***