**3GPP TSG- Meeting # *6200***

**, , - revision of S5-245868**

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

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

|  |
| --- |
|  |
| ***Title:***  |  |
|  |  |
| ***Source to WG:*** |  |
| ***Source to TSG:*** | S5 |
|  |  |
| ***Work item code:*** |  |  | ***Date:*** |  |
|  |  |  |  |  |
| ***Category:*** |  |  | ***Release:*** |  |
|  | *Use one of the following categories:****F*** *(correction)****A*** *(mirror corresponding to a change in an earlier release)****B*** *(addition of feature),* ***C*** *(functional modification of feature)****D*** *(editorial modification)*Detailed explanations of the above categories canbe found in 3GPP [TR 21.900](http://www.3gpp.org/ftp/Specs/html-info/21900.htm). | *Use one of the following releases:Rel-8 (Release 8)Rel-9 (Release 9)Rel-10 (Release 10)Rel-11 (Release 11)…Rel-17 (Release 17)Rel-18 (Release 18)Rel-19 (Release 19) Rel-20 (Release 20)* |
|  |  |
| ***Reason for change:*** | YANG SS is not matching the approved stage 2. |
|  |  |
| ***Summary of change:*** | Update YANG code to match existing stage 2. |
|  |  |
| ***Consequences if not approved:*** | Stage 2 and Stage 3 mismatch; interoperability problems. |
|  |  |
| ***Clauses affected:*** | Forge only |
|  |  |
|  | **Y** | **N** |  |  |
| ***Other specs*** |  | **X** |  Other core specifications  | TS/TR ... CR ...  |
| ***affected:*** |  | **X** |  Test specifications | TS/TR ... CR ...  |
| ***(show related CRs)*** |  | **X** |  O&M Specifications | TS/TR ... CR ...  |
|  |  |
| ***Other comments:*** | Forge MR link: <https://forge.3gpp.org/rep/sa5/MnS/-/merge_requests/1428> at commit 64be39e192dd9109ceae0213b5cbdbdf12bf848f |
|  |  |
| ***This CR's revision history:*** |  |

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

\*\*\* yang-models/\_3gpp-5g-common-yang-types.yang \*\*\*

<CODE BEGINS>

module \_3gpp-5g-common-yang-types {

 yang-version 1.1;

 namespace "urn:3gpp:sa5:\_3gpp-5g-common-yang-types";

 prefix "types5g3gpp";

 import ietf-yang-types { prefix yang; }

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

 organization "3GPP SA5";

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

 description "The model defines common types for 5G networks and

 network slicing.

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

 TTA, TTC). All rights reserved.";

 reference "3GPP TS 28.541";

 revision 2024-10-06 { reference CR-1389; }

 revision 2024-05-24 { reference CR-1273 ; }

 revision 2023-09-18 { reference CR-1043 ; }

 revision 2023-05-10 { reference CR-0916; }

 revision 2021-08-05 { reference S5-214053/CR-0518; }

 revision 2020-11-05 { reference CR-0412 ; }

 revision 2019-10-20 { reference "Initial version."; }

 grouping NcgiGrp {

 description "Represents the Ncgi datatype";

 list plmnId {

 description "This attribute represents a PLMN Identity.";

 min-elements 1;

 max-elements 1;

 key "idx";

 leaf idx { type uint32 ; }

 uses types3gpp:PLMNId ;

 }

 leaf nrCellId {

 type string;

 mandatory true;

 description "This attribute represents NR Cell Identity.

 It's a 36-bit string identifying an NR Cell Id as specified in

 clause 9.3.1.7 of TS 38.413, in hexadecimal representation. Each

 character in the string shall take a value of

 '0' to '9', 'a' to 'f' or 'A' to 'F' and shall represent 4 bits.

 The most significant character representing the 4 most significant

 bits of the Cell Id shall appear first in the string, and the

 character representing the 4 least significant bit of the

 Cell Id shall appear last in the string.

 Pattern: '^[A-Fa-f0-9]{9}$'

 Example:

 An NR Cell Id 0x225BD6007 shall be encoded as '225BD6007'.";

 }

 leaf nId {

 type string;

 mandatory true;

 description "Network Identity; Shall be present if PlmnIdNid identifies

 an SNPN (see clauses 5.30.2.3, 5.30.2.9, 6.3.4, and 6.3.8 in

 3GPP TS 23.501.";

 }

 }

 typedef NRTAC {

 type string;

 description "This holds the identity of the common Tracking Area Code

 for the PLMNs.

 allowedValues:

 a) It is the TAC or Extended-TAC.

 b) A cell can only broadcast one TAC or Extended-TAC. See TS 36.300,

 subclause 10.1.7 (PLMNID and TAC relation).

 c) TAC is defined in subclause 19.4.2.3 of 3GPP TS 23.003

 and Extended-TAC is defined in subclause 9.3.1.29 of 3GPP TS 38.473.

 d) For a 5G SA (Stand Alone), it has a non-null value.";

 }

 grouping SNssai {

 description

 "Single Network Slice Selection Assistance Information(S-NSSAI)";

 reference "3GPP TS 23.003";

 leaf sd {

 description "Slice Differentiator

 If not needed, the value can be set to ff:ff:ff.";

 type yang:hex-string {

 length 8;

 }

 reference "3GPP TS 23.003";

 }

 leaf sst {

 type uint8;

 description "Slice/Service Type.

 Values 0 to 127 belong to standardized SST range and are defined in

 3GPP TS 23.501. Values 128 to 255 belong to operator-specific range.";

 }

 }

 grouping PlmnIdNid {

 description "Represents the SCP domain specific information as defined

 in TS 29.510 ";

 uses types3gpp:PLMNId;

 leaf nid {

 type string;

 description "This attribute represents network Identity;

 Shall be present if PlmnIdNid identifies an SNPN.

 (see clauses 5.30.2.3, 5.30.2.9, 6.3.4, and 6.3.8 in TS 23.501";

 }

 }

 grouping PLMNInfo {

 description "The PLMNInfo data type define a S-NSSAI member in a specific

 PLMNId, and it have two attributes PLMNId and S-NSSAI (PLMNId, S-NSSAI).

 The PLMNId represents a data type that is comprised of mcc

 (mobile country code) and mnc (mobile network code), (See TS 23.003

 subclause 2.2 and 12.1) and S-NSSAI represents an data type, that is

 comprised of an SST (Slice/Service type) and an optional

 SD (Slice Differentiator) field";

 uses types3gpp:PLMNId;

 uses SNssai;

 }

 typedef CommModelType {

 reference "3GPP TS 23501";

 type enumeration {

 enum DIRECT\_COMMUNICATION\_WO\_NRF {

 value 0;

 description "Directly communicate to other pre-configured NF service.";

 }

 enum DIRECT\_COMMUNICATION\_WITH\_NRF {

 value 1;

 description "Directly communicate to other NF service discovered

 by NRF.";

 }

 enum INDIRECT\_COMMUNICATION\_WO\_DEDICATED\_DISCOVERY {

 value 2;

 description "Communicate to pre-configured other NF service through

 SCP as a proxy.";

 }

 enum INDIRECT\_COMMUNICATION\_WITH\_DEDICATED\_DISCOVERY {

 value 3;

 description "Communication to NF service discovered by NRF through SCP

 as a proxy.";

 }

 }

 }

 grouping CommModel {

 leaf groupId {

 type uint16;

 }

 leaf commModelType {

 type CommModelType;

 }

 leaf-list targetNFServiceList {

 type types3gpp:DistinguishedName;

 }

 leaf commModelConfiguration {

 type string;

 }

 }

 grouping SupportedFunc {

 leaf function {

 type string;

 }

 leaf policy {

 type string;

 }

 }

 typedef EnergySavingLoadThresholdT {

 type uint32 {

 range 0..10000;

 }

 units 1/10000;

 }

 typedef EnergySavingTimeDurationT {

 type uint32 {

 range 0..900;

 }

 units seconds;

 }

 typedef PhysCellID {

 type uint32 {

 range "0..1007";

 }

 reference "clause 7.4.2 of TS 38.211";

 }

 typedef UTC24TimeOfDayT {

 description "Time of day in HH:MM or H:MM 24-hour format per UTC

 time zone.";

 type string {

 pattern "(([01]?[0-9])|(2[0-3])):([0-5][0-9])";

 }

 }

 typedef DayOfWeekT {

 type enumeration {

 enum Monday;

 enum Tuesday;

 enum Wednesday;

 enum Thursday;

 enum Friday;

 enum Saturday;

 enum Sunday;

 }

 }

}

<CODE ENDS>

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

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

\*\*\* yang-models/\_3gpp-5gc-nrm-amffunction.yang \*\*\*

<CODE BEGINS>

module \_3gpp-5gc-nrm-amffunction {

 yang-version 1.1;

 namespace urn:3gpp:sa5:\_3gpp-5gc-nrm-amffunction;

 prefix amf3gpp;

 import \_3gpp-common-managed-function { prefix mf3gpp; }

 import \_3gpp-common-managed-element { prefix me3gpp; }

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

 import \_3gpp-5g-common-yang-types { prefix types5g3gpp; }

 import ietf-inet-types { prefix inet; }

 import ietf-yang-types { prefix yang; }

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

 import \_3gpp-nr-nrm-gnbcuupfunction { prefix gnbcuup3gpp ;}

 import \_3gpp-5gc-nrm-nfprofile { prefix nfp3gpp; }

 organization "3gpp SA5";

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

 description "AMFFunction derived from basic ManagedFunction.

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

 TTA, TTC). All rights reserved.";

 reference "3GPP TS 28.541 5G Network Resource Model (NRM)";

 revision 2024-10-06 { reference CR-1389; }

 revision 2024-04-04 { reference CR-1139; }

 revision 2023-09-18 { reference CR-1043 ; }

 revision 2022-01-07 { reference CR-0643; }

 revision 2020-11-06 { reference CR-0412 ; }

 revision 2019-10-25 { reference "S5-194457 S5-193518"; }

 revision 2019-05-31 { reference "Ericsson refactoring."; }

 revision 2018-08-07 { reference "Initial revision"; }

 grouping NTNGlobalRanNodeIDGrp{

 list pLMNId {

 description "The PLMN Identifier is composed of

 a Mobile Country Code (MCC) and

 a Mobile Network Code (MNC).";

 min-elements 1;

 max-elements 1;

 key "mcc mnc";

 uses types3gpp:PLMNId;

 }

 choice GlobalRanNodeID{

 description "Unique identifier of an NG-RAN node.";

 reference "3GPP TS 38.413";

 mandatory true;

 case gNB {

 leaf GlobalgNBID{

 description "Unique identifier of a gNB.";

 reference "3GPP TS 38.413";

 type int64;

 mandatory true;

 }

 }

 case ng-eNB{

 leaf Globalng-eNBID{

 type string{

 pattern '^(MacroNGeNB-[A-Fa-f0-9]{5}|LMacroNGeNB-

 [A-Fa-f0-9]{6}|SMacroNGeNB

 -[A-Fa-f0-9]{5})$';

 }

 description "This represents the identifier of

 the ng-eNB ID as specified in

 clause 9.3.1.8 of 3GPP TS 38.413.";

 mandatory true;

 }

 }

 case N3IWF{

 leaf N3IWFID{

 type string{

 pattern '^[A-Fa-f0-9]+$';

 }

 description "This IE shall contain the N3IWF

 identifier received over NGAP and

 shall be encoded as a string of

 hexadecimal characters.";

 mandatory true;

 }

 }

 case TNGF{

 leaf TNGFID{

 type string{

 pattern '^[A-Fa-f0-9]+$';

 }

 description "This represents the identifier of

 the TNGF ID as specified in clause

 9.3.1.161 of 3GPP TS 38.413 in hexadecimal

 representation.";

 mandatory true;

 }

 }

 case TWIF{

 leaf TWIFID{

 type string;

 description "This represents the TWIF identification

 as specified in clause

 9.3.1.163 of TS 38.413";

 mandatory true;

 }

 }

 case W-AGF{

 leaf W-AGFID{

 type string{

 pattern '^[A-Fa-f0-9]+$';

 }

 description "This represents the identifier of

 the W-AGF ID as specified in

 clause 9.3.1.162 of 3GPP TS 38.413

 in hexadecimal representation.";

 mandatory true;

 }

 }

 }

 }

 grouping SatelliteBackhaulInfoGrp{

 description "Defines information related to satellite backhaul category

 and corresponding information of gNB.";

 reference "3GPP TS 23.501";

 list nTNGlobalRanNodeID{

 description "Unique identifier of an NG-RAN node.";

 min-elements 1;

 max-elements 1;

 leaf GlobalRanNodeIDValue{

 type string;

 description "this relies on the choice of GlobalRanNodeID";

 }

 key "GlobalRanNodeIDValue";

 uses NTNGlobalRanNodeIDGrp;

 }

 leaf satelliteBackhaulCategory{

 description "Satellite backhaul category refers to the type of the

 satellite used in the backhaul. Only a single backhaul category

 can be indicated.";

 reference "3GPP TS 29.571";

 type enumeration{

 enum GEO;

 enum MEO;

 enum LEO;

 enum OTHER\_SAT;

 enum DYNAMIC\_GEO;

 enum DYNAMIC\_MEO;

 enum DYNAMIC\_LEO;

 enum DYNAMIC\_OTHER\_SAT;

 enum NON\_SATELLITE;

 }

 mandatory true;

 }

 leaf geoSatelliteId{

 description "Unique identifier of a GEO satellite.";

 reference "3GPP TS 29.571";

 type string;

 //condition present only if the UPF is deployed on the satellite.

 }

 }

 grouping GUAMInfoGrp {

 description "Represents the GUAMInfo datatype.";

 list pLMNId {

 description "This attribute represents a PLMN Identity.";

 min-elements 1;

 max-elements 1;

 key "idx";

 leaf idx { type uint32 ; }

 uses types3gpp:PLMNId ;

 }

 leaf aMFIdentifier {

 type int64;

 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.

 (Ref. 3GPP TS 23.003)";

 }

 }

 grouping N2InterfaceAmfInfoGrp {

 description "Represents the N2InterfaceAmfInfo datatype.";

 leaf-list ipv4EndpointAddress {

 type inet:ipv4-address;

 min-elements 1;

 description "This attribute represents available AMF endpoint

 IPv4 address(es) for N2.";

 }

 leaf-list ipv6EndpointAddress {

 type inet:ipv6-address;

 min-elements 1;

 description "This attribute represents available AMF endpoint

 IPv6 address(es) for N2.";

 }

 leaf amfName {

 type inet:host-name;

 description "This attribute represents AMF Name FQDN as defined in

 clause 28.3.2.5 of TS 23.003.";

 }

 }

 grouping BlockedLocationInfoGrp {

 description "Repreasents the datatype BlockedLocationInfo";

 list blockedLocation {

 description "This provides the geographical location at which the PLMN

 are not allowed in case of NTN.";

 min-elements 1;

 max-elements 1;

 key "mnc mcc";

 uses types3gpp:PLMNId ;

 }

 list blockedDurWindow {

 description "This provides the time durations for which the PLMN are

 not allowed at a given location in case of NTN";

 key idx;

 leaf idx { type uint32 ; }

 uses types3gpp:TimeWindowGrp;

 }

 list blockedSlice {

 description "This provides the slice for which the access is

 not allowed at a given location in case of NTN. ";

 max-elements 1;

 key idx;

 unique "sst sd";

 leaf idx { type uint32; }

 uses types5g3gpp:SNssai;

 }

 }

 grouping NTNPLMNRestrictionsInfoGrp {

 description "Represents the datatype NTNPLMNRestrictionsInfo";

 list pLMNId {

 description "This attribute represents a PLMN Identity.";

 min-elements 1;

 max-elements 1;

 key "mnc mcc";

 uses types3gpp:PLMNId ;

 }

 list blockedLocationInfoList {

 description "This defines the information related with the location

 for which the access restrictions are to be applied in case of NTN.";

 key "idx";

 leaf idx { type uint32 ; }

 uses BlockedLocationInfoGrp;

 }

 }

 grouping NtnLocationInfoGrp {

 description "Represents the NtnLocationInfo datatype.";

 list location {

 description "This defines the Location (geographical area) under

 consideration to which the satellite coverage info belongs";

 key idx;

 leaf idx { type uint32 ; }

 uses types3gpp:GeoAreaGrp;

 }

 list availabilityWindows {

 description "This attribute defines the list of time windows at which

 the satellite coverage will be available for this location.

 Either availabilityWindows or nonAvailabilityWindows shall be present.";

 key idx;

 leaf idx { type uint32 ; }

 uses types3gpp:TimeWindowGrp;

 }

 list nonAvailabilityWindows {

 description "This attribute defines the list of time windows at which

 the satellite coverage will not be available for this location.

 Either availabilityWindows or nonAvailabilityWindows shall be present.";

 key idx;

 leaf idx { type uint32 ; }

 uses types3gpp:TimeWindowGrp;

 }

 }

 grouping SatelliteCoverageInfoGrp {

 description "Represents the datatype SatelliteCoverageInfo";

 leaf nRSatelliteRATtype {

 type enumeration {

 enum NRLEO;

 enum NRMEO;

 enum NRGEO;

 enum NROTHERSAT;

 }

 description "This attribute defines the RAT Type for NR satellite

 access.";

 }

 list locationInfo {

 description "This attribute defines the information about location and

 corresponding time windows for which the satellite coverage will be

 available or unavailable.";

 key idx;

 leaf idx { type uint32 ; }

 uses NtnLocationInfoGrp;

 }

 }

 grouping SliceExpiryInfoGrp {

 description "Represents the datatype SliceExpiryInfo";

 list pLMNInfo {

 min-elements 1; // ???

 config false;

 description "It defines the PLMN(s) of a Network Function. ";

 key idx;

 leaf idx { type uint32 ; }

 uses types5g3gpp:PLMNInfo;

 }

 leaf expiryTime {

 type yang:date-and-time;

 config false;

 description "This attribute provides information about the time at

 which the slice is scheduled to be expired as it is not required

 anymore.

 This attribute will be set based on the sliceAvailability coming as

 part of ServiceProfile.";

 }

 }

 grouping MappedCellIdInfoGrp {

 description "Represents the datatype MappedCellIdInfo";

 list ntnGeoArea {

 description "This attribute indicates a specific geographical location

 mapped to Mapped Cell ID(s).";

 min-elements 1;

 max-elements 1;

 key "idx";

 leaf idx { type uint32 ; }

 uses types3gpp:GeoAreaGrp;

 }

 list mappedCellId {

 description "This attribute is in format of NCGI to indicate a fixed

 geographical area (See subclause 16.14.5 in TS 38.300";

 min-elements 1;

 max-elements 1;

 key "idx";

 leaf idx { type uint32 ; }

 uses types5g3gpp:NcgiGrp;

 }

 }

 grouping AmfInfoGrp {

 description "Represents the data type AmfInfo.";

 leaf amfRegionId {

 type int64;

 mandatory true;

 description "It represents the AMF Region ID, which identifies the

 region. AllowedValues: defined in subclause 2.10.1 of 3GPP TS 23.003 ";

 }

 leaf amfSetId {

 type int64;

 mandatory true;

 description "It represents the AMF Set ID, which is uniquely identifies

 the AMF Set within the AMF Region.

 allowedValues: defined in subclause 2.10.1 of 3GPP TS 23.003";

 }

 list taiList {

 min-elements 1;

 description "The list of TAIs.";

 key "idx";

 leaf idx { type uint32 ; }

 uses gnbcuup3gpp:TAIGrp;

 }

 list taiRangeList {

 min-elements 1;

 description "The range of TAIs.";

 key "idx";

 leaf idx { type uint32 ; }

 uses nfp3gpp:TaiRangeGrp;

 }

 list gUAMIdList {

 description "List of supported Globally Unique AMF Ids (GUAMIs).";

 config false;

 min-elements 1;

 key "idx";

 leaf idx { type uint32 ; }

 uses GUAMInfoGrp;

 }

 list backupInfoAmfFailure {

 description "List of GUAMIs for which the AMF acts as a backup for AMF

 failure.";

 min-elements 1;

 key "idx";

 leaf idx { type uint32 ; }

 uses GUAMInfoGrp;

 }

 list backupInfoAmfRemoval {

 description "List of GUAMIs for which the AMF acts as a backup for

 planned AMF removal.";

 min-elements 1;

 key "idx";

 leaf idx { type uint32 ; }

 uses GUAMInfoGrp;

 }

 list n2InterfaceAmfInfo {

 description "This attribute represents the N2 interface information of

 the AMF. ";

 max-elements 1;

 key "idx";

 leaf idx { type uint32 ; }

 uses N2InterfaceAmfInfoGrp;

 }

 leaf amfOnboardingCapability {

 type boolean;

 default false;

 description "This attribute indicates the AMF supports SNPN Onboarding

 capability. This is used for the case of Onboarding of UEs for SNPNs

 (see TS 23.501, clause 5.30.2.10).

 - FALSE: AMF does not support SNPN Onboarding;

 - TRUE: AMF supports SNPN Onboarding.";

 }

 leaf highLatencyCom {

 type boolean;

 description "This attribute indicates whether the AMF supports High

 Latency communication (e.g. for NR RedCap UE). This is used for CP NF

 to discover AMF supporting High Latency communication (see TS 23.501,

 clause 6.3.5).

 - FALSE: AMF does not support High Latency communication e.g. for

 NR RedCap UE;

 - TRUE: AMF supports High Latency communication e.g. for NR RedCap UE;";

 }

 }

 grouping AMFFunctionGrp {

 description "Represents the AMFFunction IOC";

 uses mf3gpp:ManagedFunctionGrp;

 list pLMNInfoList {

 min-elements 1;

 description "It defines the PLMN(s) of a Network Function. ";

 key idx;

 leaf idx { type uint32 ; }

 uses types5g3gpp:PLMNInfo;

 }

 container aMFIdentifier {

 description "An AMF identifier, comprising an AMF Region ID, an

 AMF Set ID and an AMF Pointer.";

 uses types3gpp:AmfIdentifier;

 }

 leaf sBIFQDN {

 description "The FQDN of the registered NF instance in the

 service-based interface.";

 type inet:domain-name;

 }

 leaf-list cNSIIdList {

 type string;

 description "It is a set of NSI ID. NSI ID is an identifier for

 identifying the Core Network part of a Network Slice instance when

 multiple Network Slice instances of the same Network Slice are deployed,

 and there is a need to differentiate between them in the 5GC.

 See NSI ID definition in clause 3.1 of TS 23.501 and

 subclause 6.1.6.2.7 of TS 29.531";

 }

 list managedNFProfile {

 key idx;

 min-elements 1;

 max-elements 1;

 uses types3gpp:ManagedNFProfile;

 description "";

 }

 list commModelList {

 min-elements 1;

 key "groupId";

 description "Specifies a list of commModel. It can be used by NF and

 NF services to interact with each other in 5G Core network ";

 reference "3GPP TS 23.501";

 uses types5g3gpp:CommModel;

 }

 list amfInfo {

 max-elements 1;

 description "This attribute represents information of an AMF

 NF Instance.";

 key "idx";

 leaf idx { type uint32 ; }

 uses AmfInfoGrp;

 }

 list nTNPLMNRestrictionsList {

 description "This attribute defines the location restrictions per PLMN

 that relates to non-terrestrial network access.";

 key "idx";

 leaf idx { type uint32 ; }

 uses NTNPLMNRestrictionsInfoGrp;

 }

 list satelliteCoverageInfoList {

 description "This attribute defines the information related to NR

 Satellite RAT type and corresponding information of satellite coverage";

 key "idx";

 leaf idx { type uint32 ; }

 uses SatelliteCoverageInfoGrp;

 }

 list sliceExpiryInfo {

 description "This provides information related to a network slice

 validity.";

 key "idx";

 leaf idx { type uint32 ; }

 uses SliceExpiryInfoGrp;

 }

 list satelliteBackhaulInfoList{

 description "Specifies a list of satellite backhaul information. It can be

 used by NF and NF services.";

 leaf GlobalRanNodeIDValue{

 type string;

 description "this relies on the choice of GlobalRanNodeID";

 }

 key "GlobalRanNodeIDValue";

 min-elements 1;

 uses SatelliteBackhaulInfoGrp;

 }

 leaf-list mdtUserConsentReqList {

 type enumeration {

 enum M1;

 enum M2;

 enum M3;

 enum M4;

 enum M5;

 enum M6;

 enum M7;

 enum M8;

 enum M9;

 enum MDT\_UE\_LOCATION;

 }

 description "represents a list of MDT measurement names that are

 subject to user consent at MDT activation.

 Any MDT measurement, whose name is not specified in this list, is not

 subject to user consent at MDT activation.";

 }

 list mappedCellIdInfoList {

 description "It provides the list of mapping between GEO area and

 Mapped Cell ID.";

 key "idx";

 leaf idx { type uint32 ; }

 uses MappedCellIdInfoGrp;

 }

 leaf aMFSetRef {

 type types3gpp:DistinguishedName;

 description "This is the DN of AMFSet";

 }

 }

 augment "/me3gpp:ManagedElement" {

 list AMFFunction {

 description "This IOC represents the AMF functionality in 5GC.

 For more information about the AMF, see 3GPP TS 23.501.

 The attribute sliceExpiryInfo is used when the validity information of

 a network slice need to be configured. The attribute

 sliceExpiryInfo.pLMNInfo indicates the network slice to which the

 validity information applies.";

 key id;

 uses top3gpp:Top\_Grp;

 container attributes {

 uses AMFFunctionGrp;

 }

 uses mf3gpp:ManagedFunctionContainedClasses;

 }

 }

}

<CODE ENDS>

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