**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 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:*** | Define a generic JobProgress data type that can be used for any providing progress and result information about any asynchronous (long running) background job. |
|  |  |
| ***Summary of change:*** | Define JobProgress data type. (YANG stage 3 only) |
|  |  |
| ***Consequences if not approved:*** | Missing YANG implementation for JobProgress data type. |
|  |  |
| ***Clauses affected:*** | 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/TR ... CR ...  |
|  |  |
| ***Other comments:*** | Dependent on stage 2 definitions in S5-221023 Forge link: <https://forge.3gpp.org/rep/sa5/MnS/-/tree/TS28.623_CR0144_Asynchronous_operation_NRM_additions_-_YANG_Stage-3> |
|  |  |
| ***This CR's revision history:*** |  |

***First change***

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

<CODE BEGINS>

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 28.623";

 revision 2022-01-03 { reference "CR-0144"; }

 revision 2021-11-01 { reference "CR-0141"; }

 revision 2021-09-30 {

 description "Added Longitude, Latitude, TenthOfDegrees, OnOff.";

 reference "CR-0138";

 }

 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.";

 }

 typedef EnabledDisabled {

 type enumeration {

 enum DISABLED ;

 enum ENABLED ;

 }

 }

 grouping ProcessMonitor {

 description "Provides attributes to monitor the progress of processes

 with specific purpose and limited lifetime running on MnS producers.

 It may be used as data type for dedicated progress monitor attributes

 when specifying the management representation of these processes.

 The attributes in this clause are defined in a generic way.

 For some attributes specialisations may be provided when specifying a

 concrete process representation.

 If a management operation on some IOCs triggers an associated

 asynchronous process (whose progess shall be monitored), this should

 also result in creating an attribute named 'processMonitor' (of type

 'ProcessMonitor') in these IOC(s). The processMonitor attribute may be

 accompanied by use-case specific additional data items.

 The progress of the process is described by the 'status' and

 'progressPercentage' attributes. Additional textual qualifications for

 the 'status' attribute may be provided by the 'progessStateInfo' and

 'resultStateInfo' attributes.

 When the process is instantiated, the 'status' is set to 'NOT\_RUNNING'

 and the 'progressPercentage' to '0'. The MnS producer decides when to

 start executing the process and to transition into the 'RUNNING' state.

 This time is captured in the 'startTime' attribute. Alternatively, the

 process may start to execute directly upon its instantiation. One

 alternative must be selected when using this data type.

 During the 'RUNNING' state the 'progressPercentage' attribute may be

 repeatedly updated. The exact semantic of this attribute is subject to

 further specialisation. The 'progessInfo' attribute may be used to

 provide additional textual information in the 'NOT\_RUNNING', 'CANCELLING'

 and 'RUNNING' states. Further specialisation of

 'progressStateInfo' may be provided where this data type is

 used.

 Upon successful completion of the process, the 'status' attribute is set

 to 'FINISHED', the 'progressPercentage' to 100%. The time is captured in

 the 'endTime' attribute. Additional textual information may be provided

 in the 'resultStateInfo' attribute. The type of

 'resultStateInfo' in this data type definition is 'String'.

 Further specialisation of 'resultStateInfo' may be provided

 where this data type is used.

 In case the process fails to complete successfully, the 'status'

 attribute is set to 'FAILED' or 'PARTIALLY\_FAILED', the current value of

 'progressPercentage' is frozen, and the time captured in 'endTime'. The

 'resultStateInfo' specifies the reason for the failure.

 Specific failure reasons may be specified where the data type defined in

 this clause is used. The exact semantic of failure may be subject for

 further specialisation as well.

 In case the process is cancelled, the 'status' attribue is first set to

 'CANCELLING' and when the process is really cancelled then to 'CANCELLED'.

 The transition to 'CANCELLED' is captured in the 'endTime' attribute.

 The value of 'progressPercentage' is frozen. Additional textual

 information may be provided in the 'resultStateInfo' attribute.

 The 'resultStateInfo' attribute is provided only for additional textual

 qualification of the states 'FINISHED', 'FAILED', 'PARTIALLY\_FAILED' or

 'CANCELLED'. It shall not be used for making the outcome, that the

 process may produce in case of success, available.

 The process may have to be completed within a certain time after its

 creation, for example because required data may not be available any

 more after a certain time, or the process outcome is needed until a

 certain time and when not provided by this time is not needed any more.

 The time until the MnS producer automatically cancels the process is

 indicated by the 'timer' attribute.";

 leaf id {

 type string;

 mandatory true;

 description "Id of the process. It is unique within a single

 multivalue attribute of type ProcessMonitor.";

 }

 leaf status {

 type enumeration {

 enum NOT\_STARTED ;

 enum RUNNING ;

 enum CANCELLING ;

 enum FINISHED ;

 enum FAILED ;

 enum PARTIALLY\_FAILED ;

 enum CANCELLED ;

 }

 config false;

 default RUNNING;

 description "Represents the status of the associated process,

 whether it fails, succeeds etc.

 It does not represent the returned values of a successfully finished

 process. ";

 }

 leaf progressPercentage {

 type uint8 {

 range 0..100;

 }

 config false;

 description "Progress of the associated process as percentage";

 }

 leaf-list progressStateInfo {

 type string;

 config false;

 description "Additional textual qualification of the states

 'NOT\_STARTED', 'CANCELLING' and 'RUNNING'.

 For specific processes, specific well-defined strings (e.g. string

 patterns or enums) may be defined as a specialisation.";

 }

 leaf resultStateInfo {

 type string;

 config false;

 description "Additional textual qualification of the states

 'FINISHED', 'FAILED', 'PARTIALLY\_FAILED and 'CANCELLED'.

 For example, in the 'FAILED' or 'PARTIALLY\_FAILED' state this

 attribute may be used to provide error reasons.

 This attribue shall not be used to make the outcome of the process

 available for retrieval, if any. For this purpose, dedicated

 attributes shall be specified when specifying the representation of

 a specific process.

 For specific processes, specific well-defined strings (e.g. string

 patterns or enums) may be defined as a specialisation.";

 }

 leaf startTime {

 type yang:date-and-time;

 config false;

 description "Start time of the associated process, i.e. the time when the

 status changed from 'NOT\_STARTED' to 'RUNNING'.";

 }

 leaf endTime {

 type yang:date-and-time;

 config false;

 description "Date and time when status changed to 'SUCCESS', 'CANCELLED',

 'FAILED' or 'PARTIALLY\_FAILED'.

 If the time is in the future, it is the estimated time

 the process will end.";

 }

 leaf timer {

 type uint32;

 units minutes;

 description "Time until the associated process is automatically cancelled.

 If set, the system decreases the timer with time. When it reaches zero

 the cancellation of the associated process is initiated by the

 MnS\_Producer.

 If not set, there is no time limit for the process.

 Once the timer is set, the consumer can not change it anymore.

 If the consumer has not set the timer the MnS Producer may set it.";

 }

 }

 typedef TenthOfDegrees {

 type uint16 {

 range 0..3600;

 }

 units "0.1 degrees";

 description "A single integral value corresponding to an angle in degrees

 between 0 and 360 with a resolution of 0.1 degrees.";

 }

 typedef Latitude {

 type decimal64 {

 fraction-digits 4;

 range "-90.0000..+90.0000";

 }

 description "Latitude values";

 }

 typedef Longitude {

 type decimal64 {

 fraction-digits 4;

 range "-180.0000..+180.0000";

 }

 description "Longitude values";

 }

 typedef OnOff {

 type enumeration {

 enum ON;

 enum OFF;

 }

 }

 // grouping ManagedNFProfile will be removed as it is

 // being moved to \_3gpp-5gc-nrm-nfprofile

 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;

 }

}

<CODE ENDS>

***End of changes***