**3GPP TSG-SA5 Meeting #143-e *S5-223388***

**Online, , 9th May 2022 - 17th May 2022**

|  |
| --- |
| *CR-Form-v12.2* |
| **CHANGE REQUEST** |
|  |
|  | **28.532** | **CR** | **0213** | **rev** | **-** | **Current version:** | **17.0.0** |  |
|  |
| *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:***  | Data change notifications YANG-in-Rest format |
|  |  |
| ***Source to WG:*** | Ericsson Hungary Ltd |
| ***Source to TSG:*** | S5 |
|  |  |
| ***Work item code:*** | TEI17 |  | ***Date:*** | 2022-04-29 |
|  |  |  |  |  |
| ***Category:*** | **F** |  | ***Release:*** | Rel-17 |
|  | *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-16 (Release 16)Rel-17 (Release 17)Rel-18 (Release 18)Rel-19 (Release 19)* |
|  |  |
| ***Reason for change:*** | The YANG\_Netconf solution set is not complete, it lacks important functionality: a method to report data changes. This document provides that functionality.Define the mapping of the notifyMOIChanges to a "YANG-in-Rest" format to be used for reporting data changes of the YANG\_Netconf solution set.This format uses the current Rest based notification with YANG based data in the path and value parameters. |
|  |  |
| ***Summary of change:*** | Define the mapping of the notifyMOIChanges to a "YANG-in-Rest" format. |
|  |  |
| ***Consequences if not approved:*** | Data change notifications are not available in the YANG\_Netconf solution set. |
|  |  |
| ***Clauses affected:*** | 12.1.3.X, 12.1.3.X.1, 12.1.3.X.2, 12.1.3.X.3, 12.1.3.X.4, 12.1.3.X.5, 12.1.3.X.5.1 |
|  |  |
|  | **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 ... S5-223359 Rel-17 CR 28.532 Correct notifyMOIChanges (stage 2) |
|  |  |
| ***Other comments:*** | After a number of offline meetings the following considerations are added:1. Producers are not likely to support both Netconf\_YANG and OpenApi solution sets in parallel. No need to update NtfSubscriptionControl to indicate the type of notification.
2. The href parameter will be encoded just as an FDQN of the producer e.g. mynode.vodafone.uk See stage 2 CR.
3. YANG based notifyMOIChanges notification will use the http header: "Content-Type: application/yang-data+json"
4. We are using the following definitions from the stage 2 CR:
* Attribute: identifies the full attribute with all its values and all its field values
* Attribute element: identifies an individual value of a multivalue attribute
* Attribute field: identifies a subpart of a structured attribute
* Attribute field element: identifies an individual value of a multivalue attribute field
* multivalue attribute: an attribute with multiplicity upper bound greater than 1
1. Addressing individual values in a multivalue attribute is by keys in YANG and in some cases index by OpenApi. YANG does not use the concept of address by index. Addressing by index also has disadvantages:
	1. Clients that don't maintain a full mirror of the provider's configuration/state will not be able to understand a change notification based on index.
	2. Clients that are temporarily out-of-sync in mirroring might misunderstand a change notification based on index.
	3. If the order of changes in notifyMOIChanges does not preserve the order of changes on the producer the notification may become incorrect
 |
|  |  |
| ***This CR's revision history:*** |  |

***First change***

#### 12.1.3.X Mapping of notifications

##### 12.1.3.X.1 Introduction

The notifications notifyMOICreation, notifyMOIDeletion, notifyMOIAttributeValueChanges should not be used in the YANG\_Netconf solution set as notifyMOIChanges provides the same functionality and more.

12.1.3.X.2 Notification notifyMOICreation

The notification is not mapped to the Netconf/YANG solution.

12.1.3.X.3 Notification notifyMOIDeletion

The notification is not mapped to the Netconf/YANG solution.

12.1.3.X.4 Notification notifyMOIAttributeValueChange

The notification is not mapped to the Netconf/YANG solution.

##### 12.1.3.X.5 Notification notifyMOIChanges

The YANG\_Netconf solution set uses the same mapping as the RESTful HTTP-based solution set as described in clause 12.1.1.2.5 with the changes and additions described below.

* Any changes reported are based on the YANG NRM definitions, even though the RESTful notification mapping is reused.
* The HTTP message of the notification will be sent with the http-header: "Content-Type: application/yang-data+json".
* The “path” shall be based on YANG addressing as defined by RESTCONF Resource Identifiers (see RFC 8040 section 3.5.3).
* The path includes the YANG module name as defined in RFC 8040.
* The "#" character before "/attributes" in "path" is not present. NETCONF/YANG does not differentiate between the stage 2 concepts of object and attribute so the "#" separator between them is not used.
* The value of "value" shall follow the JSON encoding of YANG defined in to RFC 7951.
* Attribute elements are identified by either their value (for simple attributes represented by a YANG leaf-list) or by the YANG key values otherwise; In JSON an index value is used.
* Attributes set using YANG default values, but not explicitly set by the consumer, are also reported. This may be the result of an MOI creation, or deletion of an attribute (or attribute field).
* In the rare case of an attribute or a subpart of the attribute that has a multiplicity upper bound greater then one and which is represented by a YANG keyless list, it is not possible to create, delete or replace just a single value. In such a case always the change of the complete attribute (all values) or the complete (all values, everything of that field) field is reported.

Note all following use-cases use JSON expressed in YAML notation.

Case 1: Creation of an MOI is reported with:

* operation: add
* path: YANG Resource Identifier pointing to the list entry representing the MOI
* value: a complete MOI representation, represented by the "id" node and the "attributes" container but exluding the list entry itself encoded according to RFC7951.

For example, the following instance of a "moiChanges" array item reports an object creation:

href: node1.lichtenberg.de

…

notificationId: 123456001

path: "/3gpp-common-managed-element:ManagedElement=node3/3gpp-common-measurements:PerfMetricJob=job1"

operation: add

value:

 id: job1

 3gpp-common-measurements:PerfMetricJob:attributes:

 jobId: '9865'

 fileReportingPeriod: '30'

Case 2: Deletion of an MOI is reported with:

* operation: remove
* path: YANG Resource Identifier pointing to the list entry representing the MOI
* value: not present

For example, the following instance of a "moiChanges" array item reports an object deletion:

href: node1.charlottenburg.de

…

notificationId: 123456002

path: "/3gpp-common-managed-element:ManagedElement=node3/3gpp-common-measurements:PerfMetricJob=job1"

operation: remove

Case 3: Creating a (complete) attribute is reported as follows. (Setting the value(s) of an attribute that had no value before the change):

* operation: add
* path: YANG Resource Identifier pointing to the leaf/leaf-list/container/list representing the attribute. If the attribute is represented by a list or leaf-list, then for this last data node the equal sign, the key value(s) or leaf-list value is omitted, only the list/leaf-list name shall be present
* value: the content of the leaf/leaf-list entry(s)/container/list entry(s) representing the created attribute encoded according to RFC7951. In case of attribute represented by a container/list the child data nodes are encoded only,the container/list itself is not.

For example, the following instance of a "moiChanges" array item reports setting the values of the performanceMetrics simple, multivalue attribute:

href: node1.spandau.de

…

notificationId: 123456003

path: "/3gpp-common-managed-element:ManagedElement=node3/3gpp-common-measurements:PerfMetricJob=job1/attributes/performanceMetrics"

operation: add

value:

- inOctets

- inPackets

- outPackets

Case 4: Deleting all values of a complete attribute is reported with

* operation: remove
* path: Same as in case 3.
* value: not present

For example, the following instance of a "moiChanges" array item reports deleting all values of the performanceMetrics attribute:

href: node1.pankow.de

…

notificationId: 123456004

path: "/3gpp-common-managed-element:ManagedElement=node3/3gpp-common-measurements:PerfMetricJob=job1/attributes/performanceMetrics"

operation: remove

Case 5: Replacing a (complete) attribute is reported as follows. (Removing all previous values of the attribute and setting new value(s)):

* operation: replace
* path: Same as in case 3.
* value: Same as in case 3.

Case 6: Adding a new value to a multivalue attribute (an attribute with multiplicity upper bound greater than 1) is reported as follows. (This does not imply any change to exisiting values):

* operation: add
* path: YANG Resource Identifier pointing to a leaf-list/list entry representing an attribute element(value). In case of adding a new element to an attribute with the property isOrdered=True the new element/value is inserted before the pointed element(value), unless the "insert" subparameter specifies differently.
* value: the leaf-list/list entry representing the new attribute value encoded according to RFC7951. In case of a list the child data nodes are encoded the list-entry itself is not.
* insert: an additional input subparameter is added to the moiChange input parameter. This indicates whether the new element/value was added before or after the element/value specified in path. The subparameter is only valid in case of attributes with the property isOrdered=True. It can take the values "before", "after". If missing it defaults to "before".

For example, the following instance of a "moiChanges" array item reports adding a new element/value to the "performanceMetrics" attribute before the outPackets element.:

notificationId: 123456006

path: "/3gpp-common-managed-element:ManagedElement=node3/3gpp-common-measurements:PerfMetricJob=job1/attributes/performanceMetrics/performanceMetrics=outPackets"

operation: add

insert: before

value: outOctets

Case 7: Deleting a single element/value from a multivalue attribute is reported as follows. (This does not imply any change to any other elements):

* operation: remove
* path: Same as case 6
* value: not present

Case 8: Replacement of a single value for a multivalue attribute is reported as follows. (This implies removing the old value; in case of a structured attribute removal all its subparts. This does not imply any change to any other values):

* operation: replace
* path: Same as case 6
* value: Same as case 6

For example, the following instance of a "moiChanges" array item reports replacing an element/value of the "thresholdInfoList" structured attribute:

notificationId: 123456008

path: 3gpp-common-managed-element:ManagedElement=node3/3gpp-common-measurements:ThresholdMonitor=job1/attributes/thresholdInfoList=thr1

operation: replace

value:

- idx: thr1

 thresholdDirection: UP

 thresholdValue: '4.5'

Case 9: Adding afield (subpart) of an attribute value is reported as follows (only used for structured attributes represented by a list or container in YANG):

* operation: add
* path: YANG Resource Identifier pointing to the leaf/leaf-list/container/list representing the attribute field. If the attribute field is represented by a list or leaf-list, the field has multiplicity upper bound greater than 1, with the property isOrdered=True the new element/value is inserted before the pointed element(value), unless the "insert" subparameter specifies differently.
* value: the leaf/leaf-list/container/list representing the new attribute field values encoded according to RFC7951. In case of a list/container representing the attribute field, value shall contain only the child data nodes, but not the container/list-entry itself.
* insert: In case the field has multiplicity upper bound greater than 1 and has the property isOrdered=True, the subparameter is used similarly as in case 6.

For example, the following instance of a "moiChanges" array item reports adding a value to the " hysteresis " attribute subpart:

notificationId: 123456009

path: 3gpp-common-managed-element:ManagedElement=node3/3gpp-common-measurements:ThresholdMonitor=job1/attributes/thresholdInfoList=thr1/hysteresis

operation: add

value: '10'

Case 10: Deleting a field (subpart) of an attribute is reported as follows. (only used for structured attributes represented by a list or container in YANG):

* operation: remove
* path: Same as case 9
* value: Not present

For example, the following instance of a "moiChanges" array item reports deleting all values of the "hysteresis" attribute field:

notificationId: 123456010

path: 3gpp-common-managed-element:ManagedElement=node3/3gpp-common-measurements:ThresholdMonitor=job1/attributes/thresholdInfoList=thr1/hysteresis

operation: remove

Case 11: Replacement of a field (subpart) of an attribute is reported as follows. This implies removing previous value(s). (only used for structured attributes represented by a list or container in YANG):

* operation: replace
* path: Same as case 9
* value: Same as case 9

***End of changes***