**3GPP TSG- Meeting # *2546***

**, , -**

|  |
| --- |
| *CR-Form-v12.1* |
| **CHANGE REQUEST** |
|  |
|  |  | **CR** | **draftCR** | **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:*** |  TEI17 |  | ***Date:*** | 2022-03-25 |
|  |  |  |  |  |
| ***Category:*** | F |  | ***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:*** | 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.Some stage 2 changes are also included to clarify deleting an attribute (removing all values for an attribute which allows zero as multiplicity). |
|  |  |
| ***Consequences if not approved:*** | Data change notifications are not available in the YANG\_Netconf solution set. |
|  |  |
| ***Clauses affected:*** | 11.1.1.11.2, 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 |
|  |  |
|  | **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:*** | See also S5-221039 as background information.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 receiver can determine the type of notification (Netconf\_YANG v. OpenApi) from the format of the path paramater.
3. The path parameter does not need to be globally unique only locally for the producer. The notifyMOIChanges notification already contains a systemDN parameter that identifies the producer.
4. Addressing individual values in a multivalue attribute is by keys in YANG and 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
	4. Addressing by index has dangers if multiple clients interact concurrently.
5. The path format is based on YANG RestConf resource-identifiers (without the http or the /restconf/data preamble). See [rfc8040#section-3.5.3](https://datatracker.ietf.org/doc/html/rfc8040%23section-3.5.3).
6. In cases where the MOI is the addressed entity, the top level data node in the value parameter will be the attributes container.
7. In a consumer handling both YANG and OpenApi based producers mapping between OpenApi style address and the proposed YANG style addresses is possible.
	1. YANG module names need to be derived form knowledge of the YANG models
	2. For multivalue attributes tanslation between keys and indexes can be done if the client keeps an up-to-date mirror of the producers NRMs instances. If the mirror is not available or out-of-sync the information conveyed by indexes is unreliable.
 |
|  |  |
| ***This CR's revision history:*** |  |

***First change***

11.1.1.11.2 Input parameters

|  |  |  |  |
| --- | --- | --- | --- |
| **Parameter Name** | **S** | **Information Type / Legal Values** | **Comment** |
| objectClass | M | See clause 11.1.1.7.2 | This parameter specifies the class name of the local root in the MIB |
| objectInstance | M | See clause 11.1.1.7.2 | This parameter specifies the instance of the local root in the MIB |
| notificationId | M | See clause 11.1.1.7.2 | See clause 11.1.1.7.2 |
| notificationType | M | const string “notifyMOIChanges” | See clause 11.1.1.7.2 |
| eventTime | M | See clause 11.1.1.7.2 | See clause 11.1.1.7.2 |
| systemDN | M | See clause 11.1.1.7.2 | See clause 11.1.1.7.2 |
| mOIChanges | M | SEQUENCE OF SET { notificationId (M), correlatedNotifications (O), additionalText (O), sourceIndicator (O), path (M), operation (M), value (CM)} | This parameter describes the NRM updates to be reported.The notificationId is an identifier of one MOI change.The path specifies the MOI created or deleted, or the MOI with replaced attribute values. The path may also identify an attribute or parts of an attribute in case the attribute is a structured data type.The operation specifies the type of operation that has been applied to the MOI specified by the path. It can have the values "CREATE", "DELETE" and "REPLACE". "CREATE" and "DELETE" refers to a MOI creation or deletion, respectively. "DELETE" may also refer to the deletion of an attribute or part of an attribute. "REPLACE" refers to the replacement of a complete attribute value of an existing MOI, or parts thereof in case the attribute is a structured data type.When a MOI creation is reported, the value carries an (optional) complete MOI representation.When a MOI deletion is reported, the value carries an (optional) complete MOI representation.When the replacement of the value of one or more attributes of a MOI is reported, the value carries the MOI representation without the attributes not changed.When the replacement of the value of a part of an attribute is reported, the value carries the new value of that part.When arrays are modified (by e.g. adding an array item, removing an array item or replacing an array item) the complete array shall be included in value.The reported MOI changes is an ordered list, since the creation of parent objects needs to be reported before the creation of child objects, and, vice versa, the deletion of child objects needs to be reported before the deletion of parent objects. |

***Next 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.

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 (part).

"value" shall be mapped to a JSON snippet according to RFC 7951.

The “path” shall be based on YANG addressing as defined by RESTCONF Resource Identifiers (see RFC 8040 section 3.5.3)

Case 1: Creation of an MOI is reported with:

* operation: CREATE
* path: YANG Resource Identifier pointing to the list entry representing the MOI
* value: a complete MOI representation. The class-name and id attribute is included in the path, so the value only includes the attributes container of the MOI encoded according to RFC7951.

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

{

 "notificationId": 123456789,

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

 "operation": "CREATE",

 "value": {

 "3gpp-common-measurements:PerfMetricJob":attributes": {

 "jobId": "9865",

 "fileReportingPeriod": "30"

 }

 }

}

Case 2: Deletion of an MOI is reported with:

* operation: DELETE
* 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:

{

 "notificationId": 123456789,

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

 "operation": "DELETE"

}

Case 3: Replacement of the value of one or more (complete) attributes of a MOI is reported as follows. This is applicable for attributes with a multiplicity “0..1”, “1”, or multiplicity greater than one. The operation removes all existing values and replaces them with the values provided in the “value” parameter:

* operation: REPLACE
* path: YANG Resource Identifier pointing to the list entry representing the MOI
* value: carries the MOI representation excluding unchanged attributes. The class-name and id attribute is included in the path, so the value only includes the attributes container of the MOI encoded according to RFC7951.

When creating or changing a multivalue attribute is reported (represented either by a YANG list or leaf-list), e.g. adding, removing, replacing or moving individual items, the complete multivalue attribute shall be included in value. When changing a multivalue attribute, attribute values not listed in "value" are deleted.

For example, the following instance of a "moiChanges" array item reports creating the granularityPeriod attribute and changing the fileReportingPeriod attribute:

{

 "notificationId": 123456789,

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

 "operation": "REPLACE",

 "value": {

 "3gpp-common-measurements:PerfMetricJob":attributes": {

 "granularityPeriod": "30",

 "fileReportingPeriod": "90"

 }

 }

}

Case 4: Deleting a complete attribute is reported with (removing all values of an attribute):

* operation: DELETE
* path: YANG Resource Identifier pointing to the attribute

When deleting a multivalue attribute is reported (represented either by a YANG list or leaf-list), the last Xpath predicate(s) representing the leaf-list value or the key(s) to the list entries are omitted

* value: not present

For example, the following instance of a "moiChanges" array item reports deletion of all values in the objectInstances multivalue attribute:

{

 "notificationId": 123456789,

 "path": "/3gpp-common-managed-element:ManagedElement=node3/\_3gpp-common-measurements:PerfMetricJob=job1/attributes/objectInstances",

 "operation": "DELETE"

}

Case 5: Creation or replacement of part of an attribute value reported as follows (only used for structured data types represented by a list or container in YANG) :

* operation: REPLACE
* path: YANG Resource Identifier pointing to the attribute part. For multivalue attributes the individual value shall be addressed. (Other individual values are unchanged.)
* value: carries value of the attribute part. This value will replace the complete value pointed by path, which might result in deletion of some sub-parts.

For example, the following instance of a "moiChanges" array item reports the change of the "hysteresis" sub-attribute of the multivalue attribute "thresholdInfoList":

{

 "notificationId": 123456789,

 "path": “\_3gpp-common-managed-element:ManagedElement=node3/\_3gpp-common-measurements:ThresholdMonitor=job1/attributes/thresholdInfoList=thr1/hysteresis",

 "operation": "REPLACE",

 "value": "191"

}

Case 6: Deleting part of an attribute value is reported as follows (only used for structured data types represented by a list or container in YANG):

* operation: DELETE
* path: YANG Resource Identifier pointing to the attribute or part of an attribute. For multivalue attributes the individual value shall be addressed. (Other individual values are unchanged.)
* value: not present

For example, the following instance of a "moiChanges" array item reports deletion of the "hysteresis" sub- attribute of the multivalue attribute "thresholdInfoList":

{

 "notificationId": 123456789,

 "path": “\_3gpp-common-managed-element:ManagedElement=node3/\_3gpp-common-measurements:ThresholdMonitor=job1/attributes/thresholdInfoList=thr1/hysteresis",

 "operation": "DELETE"

}

***End of changes***