**3GPP TSG-SA5 Meeting #132e *S5-*** ***204651***

**Online, 17th 28th August 2020**

|  |
| --- |
| *CR-Form-v12.0* |
| **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:***  | Update ThresholdMonitor YANG definition |
|  |  |
| ***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-12 (Release 12)**Rel-13 (Release 13)Rel-14 (Release 14)Rel-15 (Release 15)Rel-16 (Release 16)* |
|  |  |
| ***Reason for change:*** | Update YANG SS following the Stage 2 updates S5-204585 |
|  |  |
| ***Summary of change:*** | \_3gpp-common-measurements.yang |
|  |  |
| ***Consequences if not approved:*** | ThresholdMonitor IOC different in Stage 2 and YANG SS |
|  |  |
| ***Clauses affected:*** | D.2.4 |
|  |  |
|  | **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:*** |  |
|  |  |
| ***This CR's revision history:*** |  |

|  |
| --- |
| **1st 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 2020-09-03 { 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 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 "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 theshold 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 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.

 The attribute threshodInfoList defines the performance metrics to be

 monitored and the attribute monitorGranularityPeriod defines the

 granularity period to be applied.

 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 metrics in a

 single monitor. Each threshold is identified with a key called

 threshold level. A threshold is defined using the thresholdValue and

 hysteresis attributes.

 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 theshold is triggered when the high threshold value is

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

 theshold 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 ;

 }

 }

 }

}

|  |
| --- |
| **End of Change** |