**3GPP TSG-SA5 Meeting #162 *S5-253670***

Goteborg, Sweden, 25 - 29 August 2025

|  |
| --- |
| *CR-Form-v12.3* |
| **CHANGE REQUEST** |
|  |
|  | **32.423** | **CR** | **0213** | **rev** | **1** | **Current version:** | **19.3.0** |  |
|  |
| *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 |  |

|  |
| --- |
|  |
| ***Title:***  | Rel-19 CR TS 32.423 Add additional TRSR on Trace Record Header for continuous MDT |
|  |  |
| ***Source to WG:*** | CATT |
| ***Source to TSG:*** | S5 |
|  |  |
| ***Work item code:*** | TraceQoE\_OAM |  | ***Date:*** | 2025-08-15 |
|  |  |  |  |  |
| ***Category:*** | B |  | ***Release:*** | Rel-19 |
|  | *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:*** | This CR propose to enhancement on the Trace Record Header with associatedTraceRecordingSessionReference. This is used to ensure trace correlation and address security considerations for continuous MDT. |
|  |  |
| ***Summary of change:*** | Enhancement on Trace Record Header. |
|  |  |
| ***Consequences if not approved:*** | Missing support to trace record correlation for continuous MDT. |
|  |  |
| ***Clauses affected:*** | 5.2.2, G.2 |
|  |  |
|  | **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:*** |  |

\*\*\* START OF NEXT CHANGE \*\*\*

### 5.2.2 Trace Record Header

The trace record header contains the common fields as specified in the Table 5.2.2-1, in addition it may also contain vendor specific extensions.

Table 5.2.2.1 : Common fields in the trace record header

|  |  |
| --- | --- |
| Trace Record Header field name | Description |
| timeStamp (M) | Time stamp (in milliseconds since Epoch) of when the streaming trace record is produced internally in the Producer encoded as (64 bit integer) |
| nfInstanceId (M) | Unique id of the Producer NF instance that produced this trace record represented by a (String) |
| nfType (M) | Type of the Producer NF that produced this trace record represented by a (String) |
| traceReference (M) | Trace Reference (see clause 5.6 of 3GPP TS 32.422 [3]) (represented by a 6 bytes octet string) See Note 6. |
| traceRecordingSessionReference (M) | Trace Recording Session Reference (see clause 5.7 of 3GPP TS 32.422 [3]) represented by a (2 byte octet string. See Note 1.) |
| traceRecordTypeId (M) | Identifier of the trace record type (see clause 5.2.4 for details) represented by an ENUM with the following values: NORMAL,TRACE\_SESSION\_START, TRACE\_SESSION\_STOP, TRACE\_RECORDING\_SESSION\_START, TRACE\_RECORDING\_SESSION\_STOP, TRACE\_STREAM\_HEARTBEAT,TRACE\_RECORDING\_SESSION\_NOT\_STARTED, TRACE\_RECORDING\_SESSION\_DROPPED\_EVENTS,TRACE\_FILE\_OPEN,TRACE\_FILE\_CLOSE,TRACE\_FILE\_ABNORMAL\_CLOSED,TRACE\_RECORDING\_SESSION\_THROTTLED\_START,TRACE\_RECORDING\_SESSION\_THROTTLED\_STOP.(See Note 2). |
| ranUeId (O) | RAN defined id to represent a UE (8 byte octet string. See Note 3.) |
| payloadSchemaURI (O) | URI identifying the schema to be used in order to decode the payload represented by a (String. See Note 4.) |
| globalGnbId (CM) | Global gNB ID, as defined in subclause 9.3.1.6 of 3GPP TS 38.413 [23]. Applied for trace reported by gNB-CU-CP, gNB-CU-UP, gNB-DU.  |
| vendorExtension (O) | Vendor-specific extension(s) (See Note 5.) |
|  |  |
| associatedTraceRecordingSessionReference (O) | Associated Trace Recording Session Reference used in continuous MDT represented by a 2 byte octet string. |
| CHOICE 1.1: traceRecordName (O) | A unique name string of a Trace Record. This attribute is not required for Administration events.  |
| CHOICE 1.2: traceRecordId (O) | A unique integer ID of a Trace Record.  |
| NOTE 1: The *traceRecordingSessionReference* must be present for the Streaming Trace Records with non-zero size payload where the payload carries data captured for a Trace Recording Session and in administrative messages related to a Trace Recording Session (e.g. "Trace Recording Session Start" or "Trace Recording Session Stop").NOTE 2: The *traceRecordTypeId* with value "NORMAL" is used for Streaming Trace Records that do not carry an administrative message.NOTE 3: The *ranUeId* field is present in the trace record header when the identifier is supported by RAN. If RAN UE Id (see 3GPP TS 37.483 [46] and 3GPP TS 38.473 [26]) has been captured in the traced signaling messages that value is used.NOTE 4: The *payloadSchemaURI* is not required for Streaming Trace Records with payload of zero-size, or payload using common payload format (e.g. used to convey Streaming Trace administrative messages).NOTE 5: The *vendorExtension* is typically a generic list of key-value pairs.NOTE 6: The encoding of the Trace Reference is a 6-byte Octet String in BCD format. The first 3-byte Octet String is the PLMN ID which consists of MCC and MNC. The next 3-byte Octet String is the Trace ID. The PLMN ID is encoded as specified in clause 9.3.3.5 of 3GPP TS 38.413 [23]. |

\*\*\* NEXT CHANGE \*\*\*

# G.2 Trace Record Protocol Buffer (GPB) definitions

Normative GPB Trace Record schema, defined per clause 5.2:

syntax = “proto3”;

/\* Trace Record per 3GPP 32.423 specification.

 \* v16

 \*/

enum TraceRecordType {

    NORMAL = 0;

    TRACE\_SESSION\_START = 1;

    TRACE\_SESSION\_STOP = 2;

    TRACE\_RECORDING\_SESSION\_START = 3;

    TRACE\_RECORDING\_SESSION\_STOP = 4;

    TRACE\_STREAM\_HEARTBEAT = 5;

    TRACE\_RECORDING\_SESSION\_DROPPED\_EVENTS = 6;

    TRACE\_RECORDING\_SESSION\_NOT\_STARTED = 7;

 TRACE\_FILE\_OPEN = 8;

    TRACE\_FILE\_CLOSE = 9;

    TRACE\_FILE\_ABNORMAL\_CLOSED = 10;

TRACE\_RECORDING\_SESSION\_THROTTLED\_START = 11;

TRACE\_RECORDING\_SESSION\_THROTTLED\_STOP = 12;

TRACE\_SESSION\_NOT\_STARTED = 13;

}

message GlobalGnbId {

    bytes plmn\_identity = 1;

    int64 gnb\_id = 2;

}

message TraceRecordHeader {

 int64 time\_stamp = 1;

 string nf\_instance\_id = 2;

 string nf\_type = 3;

 bytes trace\_reference = 4;

 bytes trace\_recording\_session\_ref = 5;

 TraceRecordType trace\_rec\_type\_id = 6;

 optional bytes ran\_ue\_id = 7;

 optional string payload\_schema\_uri = 8;

  GlobalGnbId global\_gnb\_id = 9;

 map<string, string> vendor\_extension = 10;

 optional bytes associated\_trace\_recording\_session\_ref = x;

}

message TraceSessionStart {

  map<string, string> vendor\_extension = 1;

}

message TraceSessionStop {

  map<string, string> vendor\_extension = 1;

}

message TraceRecordingSessionStart {

map<string, string> vendor\_extension = 1;

}

message TraceRecordingSessionStop {

 string reason = 2;

  map<string, string> vendor\_extension = 1;

}

message TraceStreamHeartbeat {

  map<string, string> vendor\_extension = 1;

}

message TraceRecordingSessionDroppedEvents {

  int64 number\_of\_dropped\_events = 1;

  map<string, string> vendor\_extension = 2;

}

message TraceRecordingSessionNotStarted {

 string reason = 1;

  map<string, string> vendor\_extension = 2;

}

message TraceFileOpen {

map<string, string> vendor\_extension = 1;

}

message TraceFileClose {

map<string, string> vendor\_extension = 1;

}

message TraceFileAbnormalClosed {

 string reason = 1;

  map<string, string> vendor\_extension = 2;

}

message TraceRecordingSessionThrottledStart {

 string reason = 1;

  map<string, string> vendor\_extension = 2;

}

message TraceRecordingSessionThrottledStop {

  map<string, string> vendor\_extension = 1;

}

message TraceSessionNotStarted {

 string reason = 1;

  map<string, string> vendor\_extension = 2;

}

message CommonTracePayload {

  oneof record\_payload {

    TraceSessionStart trace\_session\_start = 1;

    TraceSessionStop trace\_session\_stop = 2;

    TraceRecordingSessionStart trace\_recording\_session\_start = 3;

    TraceRecordingSessionStop trace\_recording\_session\_stop = 4;

    TraceStreamHeartbeat trace\_stream\_heartbeat = 5;

    TraceRecordingSessionDroppedEvents trace\_recording\_session\_dropped\_events = 6;

    TraceRecordingSessionNotStarted trace\_recording\_session\_not\_started = 7;

  TraceFileOpen trace\_file\_open = 8;

    TraceFileClose trace\_file\_close = 9;

    TraceFileAbnormalClosed trace\_file\_abnormal\_closed = 10;

TraceRecordingSessionThrottledStart trace\_recording\_session\_throttled\_start = 11;

 TraceRecordingSessionThrottledStop trace\_recording\_session\_throttled\_stop = 12;

 TraceSessionNotStarted trace\_session\_not\_started = 13;

 }

}

message TraceRecordPayload {

 optional int64 payload\_size = 1;

 bytes binary\_payload = 2;

}

message TraceRecord {

 TraceRecordHeader header = 1;

 TraceRecordPayload payload = 2;

}

message StreamingTraceRecord {

 TraceRecord record = 1;

 optional CommonTracePayload administrative\_message = 2;

}