**3GPP TSG Meeting #135-e *TDoc S5-211086***

electronic meeting, online, 25 January - 3 February 2021

|  |
| --- |
| *CR-Form-v12.0* |
| **CHANGE REQUEST** |
|  |
|  | **32.423** | **CR** | 0121 | **rev** | **1** | **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:***  | Add new parameters for trace record header |
|  |  |
| ***Source to WG:*** | Ericsson |
| ***Source to TSG:*** | S5 |
|  |  |
| ***Work item code:*** | e\_5GMDT |  | ***Date:*** | 2021-01-25 |
|  |  |  |  |  |
| ***Category:*** | C |  | ***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-15 (Release 15)* *Rel-16 (Release 16)* *Rel-17 (Release 17)Rel-18 (Release 18)* |
|  |  |
| ***Reason for change:*** | Add new parameters for trace record header to make decoding process more efficient  |
|  |  |
| ***Summary of change:*** | * Added three new parameters for trace record header in 5.2.2 and G.2
* Move two parameters traceReference and traceRecordingSessionReference into new introduced parameter.
 |
|  |  |
| ***Consequences if not approved:*** |  Trace record header would not be efficient for decoding process |
|  |  |
| ***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:*** |  |

***First 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 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) |
| traceReferenceInfo (M) | Identifier of trace reference (3 byte octet string) and trace recording session reference (2 byte octet string) (see clause 5.6 and 5.7 of TS 32.422 [23]). The attribute is allowed to have multiple instances for trace record. |
|  |  |
|  |  |
| traceRecordTypeId (M) | Identifier of the trace record type (see clause 5.2.4 for details) represented by an ENUM with the following values: NORMALTRACE\_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(See Note 2). |
| ranUeId (O) | RAN defined UE Id (see 3GPP TS 38.463 [25] and 38.473 [26]) represented as of the 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.) |
| vendorTraceRecordId (O) | Vendor specific unique identifier for the trace record |
| globalGnbId (CM) | Global gNB ID, as defined in [23]. Applied for trace reported by gNB-CU-CP, gNB-CU-UP, gNB-DU.  |
| vendorExtension (O) | Vendor-specific extension(s) represented by a (Arraylist of String. See Note 5.) |
| NOTE 1: The *traceRecordingSessionReference* must be present for the 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 Trace Records that do not carry an administrative message.NOTE 3: The *ranUeId* field is present in the trace record header if it has been captured in the traced signaling messages.NOTE 4: The *payloadSchemaURI* is not required for Trace Records with payload of zero-size, or payload using common payload format (e.g. used to convey Trace administrative messages).NOTE 5: The *vendorExtension* is typically a generic list of key-value pairs. |

### 5.2.3 Trace Record Payload

The streaming trace record payload carries the captured Trace data being reported by the MnS Producer to the MnS Consumer and comprises the fields defined in Table 5.2.3-1.

Table 5.2.3.1 : Fields in the trace record payload

|  |  |
| --- | --- |
| Trace Record Payload parameter name | Description |
| payloadSize (O) | Size of payload, in bytes represented by a (64 bit integer. The field may be omitted if the solution set specific encoding/decoding has its own support for indicating the size.) |
| payload (M) | Sequence of bytes representing the binary encoded data of the specific trace recordArray of bytes. See Note 1. |
| NOTE 1: For example, trace record content per clause 4 of the present document with schema indicated in the header field *payloadSchemaURI* required for decoding. |

***Next changes***

# 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;

  }

message GlobalGnbId {

    bytes plmn\_identity = 1;

    int64 gnb\_id = 2;

}

message NgRanTraceId {

 bytes trace\_reference = 1;

 bytes trace\_recording\_session\_ref = 2;

}

message TraceRecordHeader {

 int64 time\_stamp = 1;

 string nf\_instance\_id = 2;

 string nf\_type = 3;

 repeated traceReferenceInfo trace\_reference\_info =4

 TraceRecordType trace\_rec\_type\_id = 5;

 bytes ran\_ue\_id = 6;

 string payload\_schema\_uri = 7;

 int64 vendor\_trace\_record\_id = 8;

  GlobalGnbId global\_gnb\_id = 9;

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

}

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 {

  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 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;

 }

}

message TraceRecord {

 TraceRecordHeader header = 1;

 bytespayload = 2;

}

***End of changes***