**3GPP TSG SA WG5 Meeting 134-e *6076***

**electronic meeting, online, 16th - 25th November 2020**

|  |
| --- |
| *CR-Form-v12.0* |
| **CHANGE REQUEST** |
|  |
|  | **32.423** | **CR** | 0119 | **rev** | **1** | **Current version:** | 16.2.0 |  |
|  |
| *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 GPB trace record for file based support  |
|  |  |
| ***Source to WG:*** | Ericsson |
| ***Source to TSG:*** | S5 |
|  |  |
| ***Work item code:*** | GPB\_FILE |  | ***Date:*** | 2020-11-16 |
|  |  |  |  |  |
| ***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-12 (Release 12)**Rel-13 (Release 13)Rel-14 (Release 14)Rel-15 (Release 15)Rel-16 (Release 16)* |
|  |  |
| ***Reason for change:*** | Add GPB trace record for file based support  |
|  |  |
| ***Summary of change:*** | * Added three administrative message for file based trace
* Defined logical layout for trace records
* Updated GPB trace trace record schema
 |
|  |  |
| ***Consequences if not approved:*** |  File based trace record for GPB would not be supported |
|  |  |
| ***Clauses affected:*** | 1, 2, 5.1, 5.2, 5.3, Annex G |
|  |  |
|  | **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 is related to proposed new WID S5-206074 |
|  |  |
| ***This CR's revision history:*** |  |

***First change***

# 1 Scope

The present document describes Trace data definition and management. It covers the trace records content, their format and transfer across UMTS networks, EPS networks or 5GS networks. GSM Trace is outside of the scope of this specification..

The present document also describes the data definition for Minimization of Drive Tests (MDT) across UMTS networks, EPS networks or 5GS networks.

The objectives of the present document are:

- To provide the descriptions for a standard set of Trace and MDT data;

- To define the common format of trace and MDT records; and

- To define a method for the reporting of Trace and MDT results across the management interfaces.

Clause 4 details the various Trace records content, Clause 5 defines GPB trace format for NR, Annex A provides Trace and MDT report file format, Annex B provides the trace report file conventions and transfer procedure, Annex C provides the trace reporting functional architecture and Annex D provides some trace and MDT files examples, Annex G provides normative GPB trace record schema and examples.

Trace and MDT concepts and requirements are covered in TS 32.421 [2] while Trace control and configuration management are described in 3GPP TS 32.422 [3].

The definition of Trace and MDT data is intended to result in comparability of Trace and MDT data produced in a multi-vendor wireless UMTS, EPS network, and/or 5GS network.

The following is beyond the scope of the present document, and therefore the present document does not describe:

- Any notification mechanisms or IRPs for trace. Only file transfer mechanism is specified for trace data transfer;

- Any data compression mechanisms for trace data transfer;

- Any Trace capability limitations (e.g. maximum number of simultaneous traced mobiles for a given NE).

# 2 References

The following documents contain provisions, which, through reference in this text, constitute provisions of the present document.

- References are either specific (identified by date of publication, edition number, version number, etc.) or non‑specific.

- For a specific reference, subsequent revisions do not apply.

- For a non-specific reference, the latest version applies. In the case of a reference to a 3GPP document (including a GSM document), a non-specific reference implicitly refers to the latest version of that document *in the same Release as the present document*.

[1] 3GPP TS 32.101: "Telecommunication management; Principles and high level requirements".

[2] 3GPP TS 32.421: "Telecommunication management; Subscriber and equipment trace: Trace concepts and requirements."

[3] 3GPP TS 32.422: "Telecommunication management; Subscriber and equipment trace: Trace control and configuration management ".

[4] 3GPP TR 21.905: "Vocabulary for 3GPP Specifications".

[5] W3C Recommendation "Extensible Markup Language (XML) 1.0" (Second Edition, 6 October 2000) http://www.w3.org/TR/2000/REC-xml-20001006

[6] W3C Recommendation "Namespaces in XML" (14 January 1999)
http://www.w3.org/TR/1999/REC-xml-names-19990114

[7] W3C Recommendation "XML Schema Part 0: Primer" (2 May 2001)
http://www.w3.org/TR/2001/REC-xmlschema-0-20010502

[8] W3C Recommendation "XML Schema Part 1: Structures" (2 May 2001)
http://www.w3.org/TR/2001/REC-xmlschema-1-20010502

[9] W3C Recommendation "XML Schema Part 2: Datatypes" (2 May 2001)
http://www.w3.org/TR/2001/REC-xmlschema-2-20010502

[10] International Standard ISO 8601: 1988 (E) "Representations of dates and times" (1988-06-15)
http://www.iso.ch/markete/8601.pdf

[11] 3GPP TS 32.300: "Telecommunication management; Configuration Management (CM); Name convention for Managed Objects".

[12] 3GPP TS 32.622: "Telecommunication management; Configuration Management (CM); Generic network resources Integration Reference Point (IRP): Network Resource Model (NRM)".

[13] 3GPP TS 29.274: "3GPP Evolved Packet System (EPS); Evolved General Packet Radio Service (GPRS) Tunnelling Protocol for Control plane (GTPv2-C); Stage 3".

[14] 3GPP TS 29.212: "Policy and Charging Control (PCC); Reference points".

[15] 3GPP TS 29.273: "Evolved Packet System (EPS); 3GPP EPS AAA interfaces".

[16] 3GPP TS 36.413: "Evolved Universal Terrestrial Radio Access Network (E-UTRAN); S1 Application Protocol (S1AP)".

[17] 3GPP TS 36.423 "Evolved Universal Terrestrial Radio Access Network (E-UTRAN); X2 Application Protocol (X2AP)".

[18] 3GPP TS 23.501: "System Architecture for the 5G System; Stage 2".

[19] 3GPP TS 23.502: "Procedures for the 5G System; Stage 2"

[20] 3GPP TS 38.300: "NR and NG-RAN Overall Description; Stage 2".

[21] 3GPP TS 38.331: "NR; Radio Resource Control (RRC); Protocol specification".

[22] 3GPP TS 38.401: "NG-RAN; Architecture Description".

[23] 3GPP TS 38.413: "NG-RAN; NG Application Protocol (NGAP)".

[24] 3GPP TS 38.423: "NG-RAN; Xn Application Protocol (XnAP)".

[25] 3GPP TS 38.463: "NG-RAN; E1 Application Protocol (E1AP)".

[26] 3GPP TS 38.473: "NG-RAN; F1 Application Protocol (F1AP)".

[27] 3GPP TS 24.501: "Non-Access-Stratum (NAS) protocol for 5G System (5GS); Stage 3".

[28] 3GPP TS 36.331: "Evolved Universal Terrestrial Radio Access (E-UTRA); Radio Resource Control (RRC); Protocol specification".

[29] 3GPP TS 23.107: "Quality of Service (QoS) concept and architecture".

[30] 3GPP TS 25.331: "Radio Resource Control (RRC); Protocol specification".

[31] 3GPP TS 36.314: "Evolved Universal Terrestrial Radio Access (E-UTRA); Layer 2 - Measurements".

[32] 3GPP TS 37.320: "Universal Terrestrial Radio Access (UTRA) and Evolved Universal Terrestrial Radio Access (E-UTRA); Radio measurement collection for Minimization of Drive Tests (MDT); Overall description; Stage 2".

[33] 3GPP TS 36.213: "Evolved Universal Terrestrial Radio Access (E-UTRA); Physical layer procedures".

[34] 3GPP TS 36.133: "Evolved Universal Terrestrial Radio Access (E-UTRA); Requirements for support of radio resource management".

[35] 3GPP TS 38.314: "NR; layer 2 measurements ".

[36] 3GPP TS 28.552: "Management and orchestration; 5G performance measurements".

[37] 3GPP TS 38.213: "NR; Physical layer procedures for control".

[38] 3GPP TS 36.214: "Evolved Universal Terrestrial Radio Access (E-UTRA); Physical layer; Measurements".

[39] 3GPP TS 32.425: "Telecommunication management; Performance Management (PM); Performance measurements Evolved Universal Terrestrial Radio Access Network (E-UTRAN)".

[40] IETF RFC 6455: "The WebSocket Procotol".

[41] IETF RFC 7692: "Compression Extensions for WebSocket".

[42] 3GPP TS 38.215: "NR; Physical layer measurements".

[43] 3GPP TS 28.532: "Management and orchestration; Generic management services".

[X] Language Guide (Proto 3): <https://developers.google.com/protocol-buffers/docs/proto3>

***Next change***

# 5 Trace format for GPB

## 5.1 Introduction

In Trace data reporting the individual trace records are carried in payload of the transport protocol messages Figure 5.1-1 illustrates the concept. The trace format for GPB is covered for both streaming and file based trace reporting.



Figure 5.1-1: Transport of Trace Records

As depicted in the Figure 5.1-1, each protocol-specific message delivers one or more trace records from the MnS Producer to the MnS Consumer. The header of the transport protocol message is protocol-specific. It may contain protocol specific extensions and/or options related to the transport stream. The payload of the transport protocol carries one of more Trace Records. The format of the individual Trace Records is specified in clause 5.2.

The procedures related to the connection establishment and meta-data exchange between the Streaming Trace data reporting MnS Producer and MnS Consumer are out of scope of the present document and are specified in TS 28.532 [43]

## 5.2 Trace Record

### 5.2.1 Introduction

On a high level, the structure is built up of Trace Records. All information is formatted in GPB version 3 (proto3) [X].

**Logical Layout**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Delimiter | Trace Record | Delimiter | Trace Record | Delimiter | Trace Record |

The delimiter is the size of the succeeding trace records, written as a GPB variable length 32-bit integer.

The Trace Record comprises a header and payload as shown in Figure 5.2.1-1.



Figure 5.2.1.1: Trace Record

The format of the Header in Trace Record specified in the clause 5.2.2. The format of the Payload in Trace Record specified in the clause 5.2.3.

The Trace Records may be used to carry the captured Trace data being reported by the MnS Producer to the MnS Consumer or to convey various administrative messages from the MnS Producer to the MnS Consumer. These cases are further explained in clause 5.2.4. Cases where MnS Consumer may transfer data or convey administrative messages to the MnS Producer are out of scope of the present document.

### 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) |
| traceReference (M) | Trace Reference (see clause 5.6 of 3GPP TS 32.422 [23]) (represented by a 3 bytes octet string) |
| traceRecordingSessionReference (M) | Trace Recording Session Reference (see clause 5.7 of 3GPP TS 32.422 [23]) 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: 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.) |
| 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 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. |

### 5.2.4 Trace administrative messages

#### 5.2.4.1 Introduction

The following administrative messages are defined to for trace management purposes:

- Trace Session Start

- Trace Session Stop

- Trace Recording Session Start

- Trace Recording Session Stop

- Trace Stream Heartbeat (streaming only)

- Trace Recording Session Not Started

- Trace Recording Session Dropped Events

- Trace File Open (file based only)

- Trace File Close (file based only)

- Trace File Abnormal Closed (file based only)

#### 5.2.4.2 Trace Session Start administrative message

The Trace Session Start administrative message shall be used to convey the start of a Trace Session (see 3GPP TS 32.422 [3] for details). The Trace Record in this case may have zero-size payload. The value of the traceRecordTypeId field in the Trace Record Header is set to "TRACE\_SESSION\_START". The start trace session administrative message is not used for signalling based activation as there is no separate trigger for starting the session and the trace recording session.

#### 5.2.4.3 Trace Session Stop administrative message

The Trace Session Stop administrative message shall be used to convey the stop of a Trace Session (see 3GPP TS 32.422 [3] for details). The Trace Record in this case may have zero-size payload. The value of the traceRecordTypeId field in the Trace Record Header is set to "TRACE\_SESSION\_STOP".The stop trace session administrative message is not used for signalling based activation as there is no separate trigger for stoping the session and the trace recording session.

#### 5.2.4.3a Trace Recording Session Start administrative message

The Trace Recording Session Start administrative message shall be used to convey the start of a Trace Recording Session (see 3GPP TS 32.422 [3] for details). The Trace Record in this case may have zero-size payload. The value of the traceRecordTypeId field in the Streaming Trace Record Header is set to "TRACE\_ RECORDING\_SESSION\_START".

#### 5.2.4.3b Trace Recording Session Stop administrative message

The Trace Recording Session Stop administrative message shall be used to convey the stop of a Trace Recording Session (see 3GPP TS 32.422 [3] for details). The Trace Record in this case may have zero-size payload. The value of the traceRecordTypeId field in the Streaming Trace Record Header is set to "TRACE\_ RECORDING\_SESSION\_STOP".

#### 5.2.4.4 Trace Stream Heartbeat administrative message

The Trace Stream Heartbeat administrative message may be used in absence of the captured trace data and other administrative messages from the MnS Producer to the MnS Consumer. The message is intended to indicate that a streaming trace connection is alive and does not indicate whether there is an ongoing Trace Session or not.

Transport protocol level keep-alive mechanisms may be used as an alternative (e.g. use of Ping and Pong WebSocket frames in IETF RFC 6455 [40]) and are out of scope of the present document.

#### 5.2.4.5 Trace Recording Session Not Started administrative message

The Trace Recording Session Not Started administrative message shall be used to convey that a trace recording session could not be started. For example, the number of simultaneous UE traces may be limited so that UE traces are not started when this limit is reached. It includes the detailed reason as string in the payload.

#### 5.2.4.6 Trace Recording Session Dropped Events administrative message

The Trace Recording Session Dropped Events administrative message shall be used to convey the number of dropped trace records. The message provides indication that trace records are dropped from a particular trace recording session. It includes the number of trace records dropped in the payload.

5.2.4.x Trace File Open administrative message

The Trace File Open administrative message shall be used to convey that trace file is opened for trace recording at the start of ROP period. The message provides indication when a file is opened.5.2.4.y Trace File Close administrative message

The Trace File Close administrative message shall be used to convey that trace file is closed for trace recording at the end of ROP period. The message provides indication when a file is closed.

5.2.4.z Trace File Abnormal Closed administrative message

The Trace File Abnormal Closed administrative message shall be used to convey that trace file is closed abnormally. For example, the trace file is closed due to resource constraint such as out of memory.

### 5.2.5 Void

## 5.3 Void

***Next change***

Annex G (normative):
Trace Record Protocol Buffer (GPB)

# G.1 Transport Protocol Payload Format

The payload of one transport protocol message can carry one or more trace records as specified in clause 5.1. For GPB trace payload, the overall encoding format shall adhere to the following rules:

- Each trace record is encoded as a single TraceRecord GPBv3 message following the schema in clause G.2.

- Each TraceRecord message is preceded by a length field indicating the size in bytes of the following GPB message. This length field is encoded using the GPB ‘varint’ wire format.

- If the transport message payload includes multiple trace records, the length field for the next TraceRecord message shall immediately follow the preceding message.

- No extra padding (unused bytes) is allowed anywhere in the transport message payload.

NOTE: The total length of the transport message payload is assumed to be available but encoding of this value is specific to the transport protocol in use.

# 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 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;

 bytes ran\_ue\_id = 7;

 string payload\_schema\_uri = 8;

 map<string, string> vendor\_extension = 9;

}

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***