

Public Support - Support Request #10959

Describe a sample with dynamic payload

2020-04-02 12:00 - hidden

Status:	Closed	
Priority:	Normal	
Category:		
Customer:	AUDI	Product Issue Numbers:
Department:	EF	Affected Products: ADTF 3.6.3
Requester's Priority:	Normal	Platform:
Support Level:	2nd Level	Topic: ADTF::MediaDescription
Resolution:	No Customer Feedback	FAQ Links:

Description

Supportanfrage

ich habe einen Stream, der MediaSamples überträgt, die sich zwar einen gemeinsamen Header teilen, aber abhängig vom dort enthaltenen Typ und payload anders interpretiert werden müssen. Unten findet sich ein vereinfachtes Beispiel.

Wie geht man bei der Typdefinition in ADTF3 damit am besten um?

Konkreter:

- Ist es okay, wenn eine MediaDescription nur den Beginn eines MediaSamples beschreibt? Damit wäre die Größe der beschriebenen Daten kleiner als das tatsächliche MediaSample
- In ADTF2 wurde aus Performancegründen von dynamischen Beschreibungen (Größe eines Arrays abhängig von einem Wert im Header) abgeraten. Gilt das weiterhin für ADTF3?
- Sollte ich in dem Fall sicherheitshalber komplett auf MediaDescription verzichten?

Vereinfachtes Beispiel. Inhalt des Streams sind MediaSamples vom Typ tDataA oder tDataB.

```
struct tHeader
{
    uint32_t type;
    uint32_t payload; // Big Endian
};
```

```
struct tDataA
{
    tHeader header;
    uint8_t data;
}
```

```
Struct tDataB
{
    tHeader header;
    uint32_t other_data[42];
}
```

Lösung

mit den Ethernet- sowie SOME/IP-Samples in der DevTb verhält es sich ähnlich.

Für Raw-Ethernet-Samples schauts konkret so aus:

Das Sample besteht aus einem Header mit fester Struktur, definiert von der DevTb, sowie dem eigentlichen Ethernet-Frame, der je nach Protokoll ganz unterschiedlich aussehen kann.

Die Mediadescription beschreibt nur den Sample-Header detailliert - der Ethernet-Frame wird als Array beschrieben, dessen Länge im Header steht.

Nach meinem Kenntnisstand hält sich der Performance-Overhead dieser 'dynamischen' Array-Beschreibung in Grenzen, solange

sich das Array ganz am Ende des Samples befindet.

Um nochmal explizit auf deine Fragen einzugehen:

Eine Mediadescription, die nur den Anfang des Samples beschreibt, ist mMn technisch unproblematisch.

Wenn möglich rate ich aber, das ganze Sample zu beschreiben indem du hinten ein dynamisches Array in die Beschreibung aufnimmst -

dann passt auch die Länge.

Die Performance sollte dadurch nur unwesentlich sinken.

Im Zweifel ist eine partielle Mediadescription besser wie gar keine -

so können Komponenten, die auf DDL-Basis arbeiten (z.B. das Media Description Display) zumindest eingeschränkt mit dem Sample was anfangen.

```
struct tEthernetSampleHeader
```

```
{  
    tInt64 tmTimestampBeginNs = 0;  
    tInt64 tmTimestampEndNs = 0;  
    tInt32 nErrorCode = 0;  
    tUInt32 nFrameCheckSequence = 0;  
    tUInt32 nFrameSize = 0;  
    tUInt8 aFrameData[0];  
};
```

```
<structs>
```

```
<struct name="tEthernetSample" version="1" ddlversion="4.0">  
    <element name="tmTimestampBeginNs" type="tInt64" arraysize="1">  
        <deserialized alignment="1"/>  
        <serialized bytepos="0" byteorder="LE"/>  
    </element>  
    <element name="tmTimestampEndNs" type="tInt64" arraysize="1">  
        <deserialized alignment="1"/>  
        <serialized bytepos="8" byteorder="LE"/>  
    </element>  
    <element name="nErrorCode" type="tInt32" arraysize="1">  
        <deserialized alignment="1"/>  
        <serialized bytepos="16" byteorder="LE"/>  
    </element>  
    <element name="nFrameCheckSequence" type="tUInt32" arraysize="1">  
        <deserialized alignment="1"/>  
        <serialized bytepos="20" byteorder="LE"/>  
    </element>  
    <element name="nFrameSize" type="tUInt32" arraysize="1">  
        <deserialized alignment="1"/>  
        <serialized bytepos="24" byteorder="LE"/>  
    </element>  
    <element name="aFrameData" type="tUInt8" arraysize="nFrameSize">  
        <deserialized alignment="1"/>  
        <serialized bytepos="28" byteorder="LE"/>  
    </element>  
</struct>  
</structs>
```

History

#1 - 2020-04-03 06:20 - hidden

- Project changed from Public Support to 11
- Status changed from New to In Progress
- Topic set to ADTF::MediaDescription
- Customer set to AUDI
- Department set to EF

@Jens: bist du weiterhin im Audi Kontext unterwegs?

@Wolff: vll kannst du Jens eine Lösung skizzieren, du hast das ja beim Thema SOME/IP und variablen Payload ähnlich designen müssen

#2 - 2020-04-06 12:16 - hidden

Hi Jens,

mit den Ethernet- sowie SOME/IP-Samples in der DevTb verhält es sich ähnlich.

Für Raw-Ethernet-Samples schauts konkret so aus:

Das Sample besteht aus einen Header mit fester Struktur, definiert von der DevTb, sowie dem eigentlichen Ethernet-Frame, der je nach Protokoll ganz unterschiedlich aussehen kann.

Die Mediadescription beschreibt nur den Sample-Header detailliert - der Ethernet-Frame wird als Array beschrieben, dessen Länge im Header steht.

Nach meinem Kenntnisstand hält sich der Performance-Overhead dieser 'dynamischen' Array-Beschreibung in Grenzen, solange sich das Array ganz am Ende des Samples befindet.

Um nochmal explizit auf deine Fragen einzugehen:

Eine Mediadescription, die nur den Anfang des Samples beschreibt, ist mMn technisch unproblematisch.

Wenn möglich rate ich aber, das ganze Sample zu beschreiben indem du hinten ein dynamisches Array in die Beschreibung aufnimmst - dann passt auch die Länge.

Die Performance sollte dadurch nur unwesentlich sinken.

Im Zweifel ist eine partielle Mediadescription besser wie gar keine - so können Komponenten, die auf DDL-Basis arbeiten (z.B. das Media Description Display) zumindest eingeschränkt mit dem Sample was anfangen.

```
struct tEthernetSampleHeader
{
    tInt64 tmTimestampBeginNs = 0;
    tInt64 tmTimestampEndNs = 0;
    tInt32 nErrorCode = 0;
    tUInt32 nFrameCheckSequence = 0;
    tUInt32 nFrameSize = 0;
    tUInt8 aFrameData[0];
};

<structs>
  <struct name="tEthernetSample" version="1" ddlversion="4.0">
    <element name="tmTimestampBeginNs" type="tInt64" arraysize="1">
      <deserialized alignment="1"/>
      <serialized bytupos="0" byteorder="LE"/>
    </element>
    <element name="tmTimestampEndNs" type="tInt64" arraysize="1">
      <deserialized alignment="1"/>
      <serialized bytupos="8" byteorder="LE"/>
    </element>
    <element name="nErrorCode" type="tInt32" arraysize="1">
      <deserialized alignment="1"/>
      <serialized bytupos="16" byteorder="LE"/>
    </element>
    <element name="nFrameCheckSequence" type="tUInt32" arraysize="1">
      <deserialized alignment="1"/>
      <serialized bytupos="20" byteorder="LE"/>
    </element>
    <element name="nFrameSize" type="tUInt32" arraysize="1">
      <deserialized alignment="1"/>
      <serialized bytupos="24" byteorder="LE"/>
    </element>
    <element name="aFrameData" type="tUInt8" arraysize="nFrameSize">
      <deserialized alignment="1"/>
      <serialized bytupos="28" byteorder="LE"/>
    </element>
  </struct>
</structs>
```

#3 - 2020-04-06 12:18 - hidden

- Status changed from In Progress to Customer Feedback Required

#4 - 2020-04-15 17:17 - hidden

- Project changed from 11 to Public Support

- Subject changed from ADTF3 MediaDescription to Describe a sample with dynamic payload

- Description updated

- Status changed from Customer Feedback Required to To Be Closed
- Private changed from Yes to No
- Resolution set to No Customer Feedback
- Affected Products ADTF 3.6.3 added

#7 - 2020-07-07 12:49 - hidden

- Status changed from To Be Closed to Closed