

Public Support - Support Request #2704

ADTF2 tAdditionalDataInfoIndex

2018-05-22 20:20 - hidden

Status:	Closed	Product Issue Numbers: Affected Products: ADTF 2.13.3, ADTF 2.14.0 Platform: Windows 7 64bit Topic: ADTF::Streaming FAQ Links:
Priority:	Normal	
Category:		
Customer:	BOSCH	
Department:	CC-DA/ETV	
Requester's Priority:	Normal	
Support Level:	2nd Level	
Resolution:	Solved Issue	

Description

Supportanfrage

habt Ihr ein internes Beispiel oder eine kurze Anleitung wie man enum tAdditionalDataInfoIndex { MSAI_DeviceOriginalTime = 0, MSAI_Counter = 1, MSAI_InfoUserOffset = 200 } für MSAI_InfoUserOffset verwendet?

Ich möchte zu dem Bildsample eine Check Summe übertragen und möchte dafür nicht MSAI_DeviceOriginalTime missbrauchen wenn möglich.

Lösung

Du musst einfach nur wissen, dass 0 (=MSAI_DeviceOriginalTime) bis 199 (=MSAI_InfoUserOffset - 1) reserved ist, ab 200 (=MSAI_InfoUserOffset) kannst du machen was du willst.

```
ucom::cObjectPtr<IMediaSampleInfo> pSampleInfo;  
if (IS_OK(pMediaSample->GetInterface(IID_ADTF_MEDIA_SAMPLE_INFO, (tVoid**) &pSampleInfo)))  
{  
    pSampleInfo->SetInfoInt(IMediaSampleInfo::MSAI_InfoUserOffset, 0x05ADFACE);  
    pSampleInfo->SetInfoInt(IMediaSampleInfo::MSAI_InfoUserOffset + 1, 0xCAFEBABE);  
    // ...  
}
```

History

#1 - 2018-05-23 09:05 - hidden

- Status changed from New to In Progress

- Topic set to ADTF::Streaming

Hi Bane,

folgendes Snippet sollte dir dabei helfen:

```
ucom::cObjectPtr<IMediaSampleInfo> pSampleInfo;  
if (IS_OK(pMediaSample->GetInterface(IID_ADTF_MEDIA_SAMPLE_INFO, (tVoid**) &pSampleInfo)))  
{  
    tmHardwareTime = pSampleInfo->SetInfoStr(IMediaSampleInfo::MSAI_InfoUserOffset, strChecksum);  
}
```

#2 - 2018-05-23 09:05 - hidden

- Status changed from In Progress to Customer Feedback Required

#3 - 2018-05-23 14:52 - hidden

Florian Roth wrote:

Hi Bane,

folgendes Snippet sollte dir dabei helfen:

[...]

Hallo Florian,

Meine eigentliche Frage war die konkrete Bedeutung von MSAI_InfoUserOffset, und ob man das auch erweitern und beliebige Typen nutzen kann, wie z.B.:

```
tUInt32 offset = IMediaSampleInfo::MSAI_InfoUserOffset;
```

```
pSampleInfo->SetInfoInt(offset + 1, 0x05ADFACE);  
pSampleInfo->SetInfoInt(offset + 2, 0xCAFEBABE);
```

Danke im Voraus!

VG Bane

#4 - 2018-05-23 17:49 - hidden

- Status changed from Customer Feedback Required to In Progress

Puh, das ist so alter nicht dokumentierter Code, das einzige was ich beitragen kann, in der Doku steht, dass explizit diese Werte verwendet werden. So wie ich es verstehe, betrifft es vorgefertigte Stellen im Indexed File, heißt du kannst ihn zwar ändern, was aber die Interpretation/Zuordnung betrifft, habe ich meine Zweifel (ausgewertet wird das imho nirgens). Deshalb die vorgegebenen Werte.

@Martin, hast du Input ?

#5 - 2018-05-24 08:52 - hidden

Bane hat es genau richtig vermutet: Alles kleiner als IMediaSampleInfo::MSAI_InfoUserOffset ist reserviert, alles drüber steht für eigene Definitionen zur Verfügung.

Ich nehme an die Checksumme ist ein Integer, oder? In Hinblick auf ADTF3 Kompatibilität und Performance würde ich keine Strings (so wie von Flo oben angedeutet) in IMediaSampleInfo ablegen.

#6 - 2018-05-24 09:15 - hidden

Genau ein Integer benötigen wir. Was ist die empfohlene Vorgehensweise hier? Ansonsten würde ich dafür MSAI_DeviceOriginalTime missbrauchen.

#7 - 2018-05-24 10:00 - hidden

- Status changed from In Progress to Customer Feedback Required

In Hinblick auf ADTF3 Kompatibilität und Performance würde ich keine Strings (so wie von Flo oben angedeutet) in IMediaSampleInfo ablegen.

Das hab ich total überlesen... natürlich keinen String sondern Integer...

Genau ein Integer benötigen wir. Was ist die empfohlene Vorgehensweise hier?

```
ucom::cObjectPtr<IMediaSampleInfo> pSampleInfo;  
if (IS_OK(pMediaSample->GetInterface(IID_ADTF_MEDIA_SAMPLE_INFO, (tVoid**) &pSampleInfo))  
{  
    pSampleInfo->SetInfoInt(IMediaSampleInfo::MSAI_InfoUserOffset, 0x05ADFACE);  
    pSampleInfo->SetInfoInt(IMediaSampleInfo::MSAI_InfoUserOffset + 1, 0xCAFEBABE);  
    // ...  
}
```

Ansonsten würde ich dafür MSAI_DeviceOriginalTime missbrauchen.

Nein, das ist reserviert, siehe Martins Antwort.

- "Alles kleiner als IMediaSampleInfo::MSAI_InfoUserOffset ist reserviert, alles drüber steht für eigene Definitionen zur Verfügung."

Arbeitet hierfür mit >= MSAI_InfoUserOffset

@Martin: Einwände ?

#8 - 2018-05-24 11:18 - hidden

@Flo: Nein :-).

#9 - 2018-05-24 12:40 - hidden

D.h. das muss so laufen?

```
ucom: :cObjectPtr<IMediaSampleInfo> pSampleInfo;
if (IS_OK(pMediaSample->GetInterface(IID_ADTF_MEDIA_SAMPLE_INFO, (tVoid**) &pSampleInfo))
{
    pSampleInfo->SetInfoInt(IMediaSampleInfo::MSAI_InfoUserOffset + 1, 0xCAFEBABE);
    // ...
}
```

#10 - 2018-05-24 12:46 - hidden

Ja, das Code Snippet ist ja bis auf den Index gleich.

Meins startet bei MSAI_InfoUserOffset (=200=), deins bei MSAI_InfoUserOffset + 1 (=201)

-> Du musst einfach nur wissen, dass 0 (=MSAI_DeviceOriginalTime) bis 199 (=MSAI_InfoUserOffset - 1) reserved ist, ab 200 (=MSAI_InfoUserOffset) kannst du machen was du willst.

#11 - 2018-06-04 09:13 - hidden

- *Project changed from 5 to Public Support*
- *Description updated*
- *Status changed from Customer Feedback Required to To Be Closed*
- *Private changed from Yes to No*
- *Resolution set to Solved Issue*

#12 - 2018-06-04 09:14 - hidden

- *Subject changed from ADTF2 tAdditonalDataInfoIndex to ADTF2 tAdditonalDataInfoIndex*
- *Status changed from To Be Closed to Closed*