

Public Support - Support Request #11540

Process more than one stream in a single processor

2020-06-19 11:29 - hidden

| | | |
|------------------------------|--------------|--|
| Status: | Closed | Product Issue Numbers: Affected Products: ADTF File Library 0.6.1 (BETA) Platform: Topic: FileLibrary::DatProcessing FAQ Links: |
| Priority: | Normal | |
| Category: | | |
| Customer: | AUDI | |
| Department: | EF | |
| Requester's Priority: | Normal | |
| Support Level: | 3rd Level | |
| Resolution: | Solved Issue | |

Description

Supportanfrage

Ich möchte einen Prozessor schreiben, der n Streams verarbeiten kann.
Ist dies möglich? Oder kann ein DatFile Processor nur einen Stream verarbeiten?

Lösung

Ja das ist möglich, aber nicht ganz Out-of-the-box.

Am einfachsten geht es wenn Du eine Singleton-Klasse anlegst die dann von den einzelnen Processor-Instanzen (eine Pro Stream) bedatet wird. Also mach am besten in deiner Processor::open() ein forward-open auf deine Singleton Instanz und das gleiche dann bei Processor::process().

History

#1 - 2020-06-22 08:51 - hidden

- Status changed from New to In Progress
- Topic set to FileLibrary::DatProcessing
- Customer set to AUDI
- Department set to EF
- Affected Products ADTF File Library 0.6.1 (BETA) added

#2 - 2020-06-24 16:58 - hidden

- Status changed from In Progress to Customer Feedback Required

Hallo Christian,

der der Call für den Processor bezieht sich immer auf einen Stream, also eine 1:1 Beziehung mit dem output File.

Es kommt auch immer darauf an, was das Output File hergibt.

Desweiteren sollte man sich die Frage stellen, warum die Werte die man fusionieren möchte, nicht in einen Stream zusammen sind ?
Was wäre denn dein konkreter Use Case ?

Unabhängig davon wäre ein Weg, seinen Processor eine Append Funktion via Property zu geben, dann könntest du den zweiten Stream inserten, sofern es das File hergibt.

So oder so benötigst du mehrere Schritte.

Kommt aber auf deinen Use Case speziell an.

Was ich nicht weiß, wie sich das Processing mit Substreams verhält.

Damit hättest du ja schon mal die Möglichkeit, "mehrere Streams" zu mergen, aufzuzeichnen, und diese dann ggf. gemeinsam zu prozessieren.

@Martin ?

Aber unabhngig davon sollten wir noch auf den Use Case schauen.

#3 - 2020-06-29 11:22 - hidden

Hi Christian,

ja das ist möglich, aber nicht ganz Out-of-the-box.

Am einfachsten geht es wenn Du eine Singleton-Klasse anlegst die dann von den einzelnen Processor-Instanzen (eine Pro Stream) bedatet wird. Also mach am besten in deiner Processor::open() ein forward-open auf deine Singleton Instanz und das gleiche dann bei Processor::process().

Ich hoffe Du kannst mir folgen, bitte melde Dich sonst gerne wieder bei uns.

Grüße,

Martin

#4 - 2020-07-02 16:30 - hidden

Gibt es noch Anmerkungen ?

#7 - 2020-07-08 13:33 - hidden

Hallo,

also habe ich mich nun dafür Entschieden einfach n Prozessoren in n Prozessen zu starten und einfach darauf zu warten bis sie durchgelaufen sind

#8 - 2020-07-08 13:33 - hidden

Ticket kann geschlossen werden

#9 - 2020-07-09 09:50 - hidden

- *Project changed from 11 to Public Support*
- *Subject changed from ifhd adtf dattool to Process more than one stream in a single processor*
- *Description updated*
- *Status changed from Customer Feedback Required to To Be Closed*
- *Private changed from Yes to No*
- *Resolution set to Solved Issue*
- *Support Level changed from 2nd Level to 3rd Level*

#10 - 2020-10-06 10:27 - hidden

- *Status changed from To Be Closed to Closed*