Public Support - Support Request #5181

Support for harmonized UTF-8 Support

2018-12-06 18:48 - hidden

Status: Closed

Priority: Normal

Category:

Customer: DAIMLER Product Issue

Numbers:

Department: RD/FAS Affected Products: ADTF 2.13.2

Requester's Priority: Normal Platform: Windows 7 64bit

Support Level: 2nd Level Topic: ADTF::Common

Resolution: Workaround Available FAQ Links:

Description

Supportanfrage

Bei XKrug haben wir ein Problem in deren ADTF flexray signal sender gemeldet (Ticket #8542) das wiederum auf ein Fehlverhalten innerhalb von ADTF schliesen lässt.

Nachstehend die Aussage von Jürgen Bubeck von XKrug mit der Bitte um Bearbeitung:

Der vom Parser gemeldete Fehler prüft auf Existenz der Datei und verhindert einen Crash des kompletten ADTF_Programms. Dieser würde auftreten, wenn die Systembibliothek einen ungültigen Dateinamen übergeben bekommen würde.

Grund:

ADTF liefert für das Property mit dem Dateipfad einen ASCII Character Array. Die Weiterverwendung dieses Strings ist bei der Nutzung von Umlauten o.Ä. nicht möglich, d.h. die angefragte Datei ist nicht auffindbar.

Das Verhalten ist von unserer Seite nicht behebbar.

Bitte diesen Fehler beim ADTF Support melden.

Workaround:

Bis einmal eine ADTF-Version existiert, die Properties als Unicode verarbeitet, kann ASCII ohne Probleme verwendet werden.

Lösung

Ja genau. Der Configuration Editor verwendet kein UTF-8 sondern für die Umlaute letztendlich ASCII-Extended Codes. Die libc versteht aber nur ASCII und UTF-8. Wenn man die Konfigurationsdatei (system.xml) händisch in UTF-8 konvertiert, kriegen die Filter auch korrekte UTF-8 Strings und dann klappen auch die cFileSystem::Exists und cFile::Open Calls wie erwartet.

UTF-8 Support wird in den GUI Komponenten von ADTF2 nicht mehr Einzug halten, bleiben also leider nur die zwei Workarounds:

- Keine Dateinamen mit Umlauten etc. verwenden.
- system.xml manuell nach UTF-8 konvertieren.

History

#1 - 2018-12-07 07:52 - hidden

- Status changed from New to In Progress
- Topic set to ADTF::Common

@Martin: Was können wir hier in ADTF 2.x noch machen? Können wir hier in einer Patch Version noch was machen? Allerdings ist ja ein Workaround vorhanden, damit wäre es kein Grund für eine Anpassung in ADTF 2.x

#2 - 2018-12-07 09:32 - hidden

Der Workaround ist, dass der Benutzer keine Umlaute oder andere Sonderzeichen in Dateinamen nutzen darf. Könnte das nicht auf ein Problem mit dem Coding hindeuten?

ISO8859-15 oder UT8?

Vielleicht liegt es ja "nur" daran dass verschiedene Komponenten ein unterschiedliches Coding nutzen?

#3 - 2018-12-11 09:29 - hidden

2024-04-28

Ja genau. Der Configuration Editor verwendet kein UTF-8 sondern für die Umlaute letztendlich ASCII-Extended Codes. Die libc versteht aber nur ASCII und UTF-8. Wenn man die Konfigurationsdatei (system.xml) händisch in UTF-8 konvertiert, kriegen die Filter auch korrekte UTF-8 Strings und dann klappen auch die cFileSystem::Exists und cFile::Open Calls wie erwartet.

UTF-8 Support wird in den GUI Komponenten von ADTF2 nicht mehr Einzug halten, bleiben also leider nur die zwei Workarounds:

- Keine Dateinamen mit Umlauten etc. verwenden.
- system.xml manuell nach UTF-8 konvertieren.

@Flo, ich gehe zwar davon aus, dass das in ADTF3 inzwischen passt, aber das müssen wir nochmal überprüfen.

#4 - 2018-12-11 13:04 - hidden

Das Coding in der system.xml ist als ISO8859-1 in der Datei definiert.

Das Coding wird doch i.d.R. vom XML Parser vorgegeben bzw. entsprechend genutzt.

Am Ende müssen alle beteiligten Komponenten hier eine einheitliche Linie fahren und auch die Settings in der xml Datei müssen dazu passen. Würde mich wundern, wenn eine Datei die von ADTF geschrieben wird so einfach konvertiert werden kann und dann soll alles funktionieren?

Aber wie schon gesagt ist das kein dringliches Problem.

Dann müssen die Kollegen eben bis auf weiteres darauf achten keine Umlaute in den Pfaden zu nutzen.

Aber für ADTF3 wäre es natürlich wünschenswert dass das Zeichenhandling dort in Pfaden usw. passt.

#5 - 2018-12-17 09:57 - hidden

- Project changed from 9 to Public Support
- Subject changed from Nutzung von Umlauten in Dateipfaden für Filter Parameter to Support for harmonized UTF-8 Support
- Description updated
- Status changed from In Progress to To Be Closed
- Private changed from Yes to No
- Resolution set to Workaround Available

#6 - 2018-12-17 09:57 - hidden

- Status changed from To Be Closed to Closed

2024-04-28 2/2