

Public Support - Support Request #15232

Failed plugin description generation when using tensorflow

2021-09-21 07:17 - hidden

Status: Closed	
Priority: Normal	
Category:	
Customer: CARIAD	Product Issue Numbers:
Department:	Affected Products: ADTF 3.10.0
Requester's Priority: Normal	Platform: Windows 10 64bit
Support Level: 2nd Level	Topic: ADTF::PluginDescription
Resolution: Solved Issue	FAQ Links:

Description

Supportanfrage

ich möchte gerne aus einem selbstgeschriebenen ADTF 3 Filter heraus tensorflow verwenden. Zum Testen habe ich ein Minimalbeispiel erstellt: Dazu habe ich den Data Processor Filter aus dem Tutorial erstellt und lediglich drei Zeilen Code eingefügt (+ die entsprechenden Verknüpfungen zur tensorflow Bibliothek in VS 2019 gesetzt):

Zwei neue "includes":

```
#include <tensorflow/c/c_api.h>
#include <iostream>
```

und dann z.B. in `cTutorialFilterDataProcessor::cTutorialFilterDataProcessor()` folgende Zeile als Test hinzugefügt:
`std::cout << "Hello from TensorFlow C library version " << TF_Version() << std::endl;`

In einem reinen C++ Projekt (ohne ADTF Bezug) funktioniert der build problemlos. Im ADTF Filter funktioniert der Gesamtbuid der Solution nicht mehr, sobald man die Zeile mit der Benutzung von tensorflow einfügt, hier konkret der Aufruf `TF_Version()`.

Ein separater Build des `tutorial_filter_data_processor` funktioniert. Das Problem liegt beim build des Projekts `tutorial_filter_data_processor_pdgen`, ein Screenshot der leider nicht sehr aussagekräftigen Fehlermeldung habe ich angefügt.

Leider ist mir nicht ganz klar was die Rolle dieses `_pdgen`-Projekts ist und wieso hier ein Problem mit tensorflow auftreten sollte.

Ich habe auch schon versucht die tensorflow Benutzung in ein anderes Projekt auszulagern. Sobald ich dieses allerdings aus dem ADTF Filter heraus aufrufe führt es zum selben Fehler.

Gibt es vielleicht schon Erfahrung, wie tensorflow aus einem ADTF Filter heraus verwendet werden kann?

Lösung

Der Verweis zu Tensorflow hat gefehlt, das kann mittels `DEPENDENT_DYNAMIC_LIBS` definiert werden. Siehe auch:

- https://support.digitalwerk.net/adtf/v3/guides/tools_adtf_plugin_description_generator.html
- https://support.digitalwerk.net/adtf/v3/guides/sdk_generate_plugin_description.html

History

#1 - 2021-09-21 10:55 - hidden

- Status changed from New to In Progress
- Customer set to CARIAD

Hallo Talitha,

wahrscheinlich liegt es nicht an tensorflow sondern an dem VS2019. Wir können bis jetzt keine 100% Freigabe für ADTF und VS2019 geben - sind aber dran dies zu ändern.

Folgende Info habe ich noch für dich:

Currently for ADF 3.12.7, VS2017 VC141 is in use.
However you can also use VS2019 (IDE) in combination with VC141 to use new IDE features.
Theoretically VC142 (Standardtoolchain von VS2019) may be compatible, but may have cross effects with Qt (=VC141).

If you choose to use VS2019 + VC141, please be aware you will need to configure Conan + CMake accordingly.

Current ADF is compiled and tested with VS2017 VC141:
https://support.digitalwerk.net/adtf/v3/adtf_html/page_system_requirements.html#section_windows_x64_requirements
There is currently no ADF delivery for VS2019 VC142 available but planned until the following months to change to this toolchain.

Viele Grüße
Margarete

#5 - 2021-09-22 09:28 - hidden

Hallo Talitha,

könntest du uns den kompletten output vom build in Visual Studio zusenden?
Sowohl beim Bauen des Filters als auch der Plugin Description.
Welchen build type nutzt du und passt dieser zu tensor flow?

Viele Grüße
Margarete

#6 - 2021-09-22 10:04 - hidden

- File log.txt added

Hallo Margarete,

erst Mal vielen Dank für deine schnelle Antwort. Ich versuche mir gerade für VS2019 die VC141 Toolkette zu besorgen (und danach zu integrieren) um zu schauen ob das hilft. Aktuell benutze ich definitiv VC142.
Da ich außerdem ADF 3.10 verwende noch folgende Frage: ist dafür auch VS2017 VC141 vorgesehen und getestet, oder brauche ich dafür eine noch ältere Version?

Ich habe dir außerdem wie gewünscht den Output vom kompletten build angehängt.

Ich arbeite momentan lediglich im Release Modus, ich habe auch keinen Debug-Build von tensorflow.

In einem C++ Projekt ohne ADF Bezug funktioniert das Zusammenspiel mit tensorflow dementsprechend auch nur im Release Modus. Zum Ausführen (post-build) eines solchen Projekts muss dann noch eine tensorflow.dll in das richtige Verzeichnis (wo die .exe erstellt wird) kopiert werden. Hier ist mir auch noch unklar was entsprechend für das ADF Plugin gemacht werden müsste. Aber ich denke diesen Schritt müssen wir erst angehen, wenn wir den build erfolgreich abschließen.

Viele Grüße
Talitha

#8 - 2021-09-22 12:00 - hidden

Kurzes Update: Ich habe jetzt in VS2019 die Toolkette v141 nachinstalliert und für das gesamte ADF Filter Projekt darauf umgestellt. Leider bekomme ich beim build noch immer den selben Fehler.

#9 - 2021-09-23 07:32 - hidden

Ich denke ich habe das Problem gelöst: Es scheint, dass der build des _pdgen Projekt bereits die tensorflow.dll benötigt - auch wenn mir nicht klar ist wofür. D.h. wenn ich die tensorflow.dll bereits in das Verzeichnis lege wo das plugin und die plugin description erstellt werden, dann funktioniert der build und es gibt keine Fehlermeldung mehr.

Viele Grüße

#10 - 2021-09-23 08:41 - hidden

- Status changed from In Progress to To Be Closed

- Topic set to ADF::Common

- Resolution set to Solved Issue

Hallo Talitha,

danke für die Info und gut das es sich von "selbst" gelöst hat.
Damit schließen wir das Ticket.

Grüße

Margarete

#11 - 2021-09-23 08:43 - hidden

- Description updated

#12 - 2021-09-27 07:48 - hidden

- Project changed from 30 to Public Support

- Subject changed from ADTF 3 + tensorflow to Failed plugindescription generation when using tensorflow

- Description updated

- Private changed from Yes to No

- Topic changed from ADTF::Common to ADTF::PluginDescription

Hallo Talitha,

Es scheint, dass der build des _pdgen Projekt bereits die tensorflow.dll benötigt - auch wenn mir nicht klar ist wofür.

Der Plugin Description Generator lädt das Plugin und fährt es hoch um zu wissen, welche Pins, Properties usw vorhanden sind, erstellt daraus eine Plugin Description als Meta Information, damit zb der Configuration Editor die Inhalt des Plugins kennt, ohne diese Instanzieren zu müssen. D.h. der Plugin Description Generator benötigt alle Runtime Abhängigkeiten, sonst klappt das Laden natürlich nicht. Siehe https://support.digitalwerk.net/adtf/v3/guides/tools_adtf_plugin_description_generator.html

D.h. wenn ich die tensorflow.dll bereits in das Verzeichnis lege wo das plugin und die plugindescription erstellt werden, dann funktioniert der build und es gibt keine Fehlermeldung mehr.

Unter dem angegebenen Link findest du auch eine elegantere Möglichkeit tensorflow zu referenzieren, dann musst du es nicht ins Verzeichnis legen. Das ersetzt aber nicht deinen install bzw. deploy, dort muss es vermutlich auch bekannt sein, wo tensorflow zu finden ist. Dennoch ist die Verwendung von DEPENDENT_DYNAMIC_LIBS die korrekte Vorgehensweise. Bedenke aber dabei, dass du hier alle weiteren Abhängigkeiten ggf. auch angeben musst, sollte tensorflow weitere DLLs benötigen.

#13 - 2022-01-17 08:17 - hidden

- Status changed from To Be Closed to Closed

Files

ADTF_tensorflow_Fehler.PNG	25.5 KB	2021-09-21	hidden
log.txt	11.8 KB	2021-09-22	hidden