

Public Support - Support Request #11012

Using conan packages and required package _DIRs

2020-04-14 09:30 - 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::Common
Resolution:	Solved Issue	FAQ Links:
Description		
Supportanfrage		
<p>Wir steigen gerade auf ADTF3 um und wollen die ADTF (Conan) Pakete von ihrem Artifactory verwenden - https://artifactory.digitalwerk.net/artifactory/webapp</p> <p>Uns stellen sich leider ein paar Grundlegende Fragen, wie man ADTF sauber mit den Toolboxes und OSG und Qt starten kann. Wir verwenden in Conan den virtualenv und cmake_find_package Generator. Den cmake_find_package benötigen wir natürlich um die Abhängigkeiten für die Filter- und Mixinprogrammierung zu erhalten. Virtualenv nutzen wir um uns eine Laufzeitumgebung zu erstellen, aus der heraus wir dann u.a. ADTF3 starten können.</p> <p>Leider ist es unserer Meinung nach nicht möglich ADTF sauber mit allen Toolboxes zu starten, ohne viel Hand anzulegen.</p> <p>Sowohl ADTF als auch alle Toolboxes fügen sich an die Umgebungsvariabel ADTF_DIR hinzu. Das sorgt dafür das es sich um eine Liste enthält. Um ADTF zu starten bräuchte man aber einen Pfad zum bin-Ordner vom ADTF-Paket. So muss man leider suchen, wo das ADTF Paket im Conan-Cache liegt und es dann manuell starten.</p> <p>Durch die Verwendung von Conan legen die Toolboxes jeweils in eigene Ordner. Die Anleitungen gehen davon aus, dass sie im addons-Ordner von ADTF liegen. Abhilfe könnte man sich schaffen, wenn es eindeutige Umgebungsvariablen für die Toolboxes gibt, damit man diese in den Projekteigenschaften angeben kann. Die Dokumentation beschreibt das sogar, in dem z.B. folgende Variabel gesetzt sein soll - ADTF_DISPLAY_TOOLBOX_DIR. Warum wird sie dann nicht vom Conan-Rezept der Disaply-Toolbox gesetzt.</p> <p>Laut den verschiedenen Dokumentationen sollten folgende Umgebungsvariablen definiert sein.</p> <p>ADTF_DIR: Verzeichnis der ADTF installation OSG_DIR: Verzeichnis der Installation von OpenSceneGraph QT_Dir: Verzeichnis der QT Installation ADTF_DISPLAY_TOOLBOX_DIR: Verzeichnis DisplayToolbox. ADTF_DEVICE_TOOLBOX_DIR: Verzeichnis Device Toolbox</p> <p>ADTF_DIR ist gesetzt aber eben falsch da es eine Liste von ADTF und den Toolboxes ist. Alle andere Variablen werden nicht durch die Conan-Rezepte definiert.</p> <p>Wie sollte dann mit den aktuell vorhandenen Conan-Paketen eine sauber ADTF-Umgebung erzeugt werden und warum werden die Variablen nicht einfach in den Rezepten angegeben?</p>		
Lösung		
<p>siehe #9404 deines Kollegen, wir werden das Stückweise nachziehen wo es Sinn macht.</p> <p>Vielleicht noch als Ergänzung allgemein: Für ADTF existiert keine Anforderung/Kundenforumsauftrag, ein Conan Paket bereitzustellen, sondern ein Archiv. Wir "missbrauchen" Conan derzeit dazu, um diese Anforderung zu erfüllen, weil wir ebenfalls mit Conan entwickeln. Gleichzeitig können wir damit auch die Conan Anwender indirekt bedienen, ebenso stellen wir die 3rd Party SDK Pakete so wie wir sie verwenden mit bereit.</p> <p>Bisher unterstützt das den Entwicklungsfall, das läuft soweit und wir haben HelferSkripte, um die <Produkt>_DIR zu setzen. Das hilft natürlich nur bei der Entwicklung, beim Launch sieht es da wieder anders aus, das ist (bzw. war) aber auch bisher nicht der Teil der Aufgabe.</p>		

Wir werden das dennoch im Zuge unseres Deploy Prozesses nach und nach anpassen, hier besteht aber bisher kein Auftrag und läuft somit low-prior on top.
Ich verweise vorsorglich noch einmal auf die fehlende aber suggerierte Conan Anforderung (das müsste eigentlich die AEV für AUDI machen, somit geht es aktuell nur soweit es aktuell geht), ebenso auf [#9404](#).
Bei der AEV gibt es meines Wissens einen adtf starter und einen ce starter, bei den Pfade für die Runtime Umgebung gesetzt werden, eben für diesen Conan Kontext.
Ob das mittlerweile voll ausgreift ist und auch sämtliche notwendigen Variablen durchreicht, weiß ich leider an dieser Stelle nicht, überschreitet aber auch meinen Support Range.

Die Idee ist mittelfristig / spätestens zum 3.8.0 release das Deployment zu ändern.

Das soll dann wie folgt aussehen:

Conan:
Für Entwickler die gegen ADTF Entwickeln.

- First / Second Party SDK's (a_util, ddl, ...)
- Third Party SDK's (OSG, ...) <- Hier will ich aber nach Möglichkeit auch auf Pkgs aus dem Conan-Center umstellen (Wenn Erreichbarkeit aus dem Netz der Kunden gegeben ist).
- ADTF-SDK pkg <- kleine SDK Pakete (debug/release jeweils einzeln) rein zur Entwicklung, keine GUI Tools etc.
- Toolboxen

Zip-Download:
Für Anwender.

- Vollständiges ADTF mit allen GUI-Tools etc., debug und release sind in einem Archiv zusammen.
- Toolboxen, debug und release sind in einem Archiv zusammen (ein Archiv pro Toolbox).

Ansonsten müssen zum Build sämtliche CMake relevanten Variablen gesetzt werden, das lässt sich auch im conan recipe machen.

Im Falle von Mixins:

```
set (QT_DIR ${CONAN_QT_ROOT})
set (OSG_DIR ${CONAN_OSG_ROOT})

find_package (ADTF_DISPLAY_TOOLBOX REQUIRED COMPONENTS mixin)
```

Related issues:	
Related to Public Support - Support Request #9404: Conan ADTF and SDK package...	Closed

History

#1 - 2020-04-14 10:02 - hidden

- Related to Support Request #9404: Conan ADTF and SDK packages do not add their bin directories to path added

#2 - 2020-04-14 10:11 - hidden

- Project changed from Public Support to 11
- Description updated
- Status changed from New to In Progress
- Topic set to ADTF::Common
- Resolution set to Not Supported Scope
- Customer set to AUDI
- Department set to EF
- Affected Products ADTF 3.6.3 added

Hallo Hubert,

siehe [#9404](#) deines Kollegen, wir werden das Stückweise nachziehen wo es Sinn macht.

Vielleicht noch als Ergänzung allgemein:
Für ADTF existiert keine Anforderung/Kundenforumsauftrag, ein Conan Paket bereitzustellen, sondern ein Archiv.
Wir "missbrauchen" Conan derzeit dazu, um diese Anforderung zu erfüllen, weil wir ebenfalls mit Conan entwickeln.
Gleichzeitig können wir damit auch die Conan Anwender indirekt bedienen, ebenso stellen wir die 3rd Party SDK Pakete so wie wir sie verwenden mit

bereit.

Bisher unterstützt das den Entwicklungsfall, das läuft soweit und wir haben HelferSkripte, um die <Produkt>_DIR zu setzen. Das hilft natürlich nur bei der Entwicklung, beim Launch sieht es da wieder anders aus, das ist (bzw. war) aber auch bisher nicht der Teil der Aufgabe. Wir werden das dennoch im Zuge unseres Deploy Prozesses nach und nach anpassen, hier besteht aber bisher kein Auftrag und läuft somit low-prior on top.

Ich verweise vorsorglich noch einmal auf die fehlende aber suggerierte Conan Anforderung (das müsste eigentlich die AEV für AUDI machen, somit geht es aktuell nur soweit es aktuell geht), ebenso auf [#9404](#).

Bei der AEV gibt es meines Wissens einen adtf starter und einen ce starter, bei den Pfade für die Runtime Umgebung gesetzt werden, eben für diesen Conan Kontext.

Ob das mittlerweile voll ausgreift ist und auch sämtliche notwendigen Variablen durchreicht, weiß ich leider an dieser Stelle nicht, überschreitet aber auch meinen Support Range.

@Nils: Ergänzungen dazu ? Workarounds ? Zeithorizont ?

#3 - 2020-04-14 16:37 - hidden

Hallo,

gibt es dann irgendwo eine Anleitung, wie man die ganzen Archive (ADTF, ADTF-DisplayToolbox, OSG und QT) zusammenkonfigurieren muss, damit man ein Mixin samt OSG- und QT-Support kompilieren kann?

Leider haben wir nichts entsprechendes gefunden. Aktuell haben wir mit CMake eine Konfiguration startet, allerdings fehlen die Verweise auf die konkreten OSG- und QT-Libs weshalb es zu Linkerfehlern kommt.

#5 - 2020-04-14 17:49 - hidden

Hallo Hubert,

dein EF-Kollege Nico Juralewsky macht scheinbar gerade das gleiche, siehe #10996.

Oder auch Falk Pastor, mit ihm waren wir schon beisammen wegen Mixin bauen und sind im Austausch weiterer Conan Anpassungen (u.a. Package Dir im Package setzen o.ä.)

Vielleicht wollt ihr hier mal Synergien schöpfen und ein Best Practise posten ?

Eigentlich musst du nur ein Conan recipe mit den build requirements aufsetzen (ADTF, DispTB, Qt und OSG bekommst du bei uns im Artifactory), die CMake Variablen dazu gleich setzen und dich an den Mixin Examples in der DispTB orientieren (find_package etc).

Anders machen es diese Examples bzw die Toolbox auch nicht.

#6 - 2020-04-14 21:40 - hidden

- Status changed from In Progress to Customer Feedback Required

#7 - 2020-04-15 08:04 - hidden

Florian Roth wrote:

@Nils: Ergänzungen dazu ? Workarounds ? Zeithorizont ?

Hallo Hubert,

Die Idee ist mittelfristig / spätestens zum 3.8.0 release das Deployment zu ändern.

Das soll dann wie folgt aussehen:

Conan:

Für Entwickler die gegen ADTF Entwickeln.

- First / Second Party SDK's (a_util, ddl, ...)
- Third Party SDK's (OSG, ...) <- Hier will ich aber nach Möglichkeit auch auf Pkgs aus dem Conan-Center umstellen (Wenn Erreichbarkeit aus dem Netz der Kunden gegeben ist).
- ADTF-SDK pkg <- kleine SDK Pakete (debug/release jeweils einzeln) rein zur Entwicklung, keine GUI Tools etc.
- Toolboxes

Zip-Download:

Für Anwender.

- Vollständiges ADTF mit allen GUI-Tools etc., debug und release sind in einem Archiv zusammen.
- Toolboxes, debug und release sind in einem Archiv zusammen (ein Archiv pro Toolbox).

Bei Wünschen, Fragen, Ergänzungen einfach melden.

Schöne Grüße,

Nils

#8 - 2020-04-15 11:39 - hidden

Hallo,

wir konnten CMake jetzt erfolgreich konfigurieren. Es ging aber wirklich nur, in dem wir vorher in CMake die Variablen für OSG_DIR und QT_DIR gesetzt haben. Das ging sogar mit den standardmäßigen Conan-Variablen

Bsp:

```
set (QT_DIR ${CONAN_QT_ROOT})  
set (OSG_DIR ${CONAN_OSG_ROOT})  
  
find_package (ADTF_DISPLAY_TOOLBOX REQUIRED COMPONENTS mixin)
```

#9 - 2020-04-15 12:38 - hidden

- Status changed from *Customer Feedback Required* to *In Progress*

#10 - 2020-04-15 17:12 - hidden

- Project changed from *11* to *Public Support*
- Subject changed from *WG: ADTF Conan Pakete* to *Using conan packages and required package _DIRs*
- Description updated
- Status changed from *In Progress* to *To Be Closed*
- Private changed from *Yes* to *No*
- Resolution changed from *Not Supported Scope* to *Solved Issue*

#13 - 2020-07-07 12:49 - hidden

- Status changed from *To Be Closed* to *Closed*