

Public Support - Support Request #10968

ADTF3 Conan Package Variants (EB vs. DW)

2020-04-03 11:15 - 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: 3rd Level	Topic: ADTF::License
Resolution: Solved Issue	FAQ Links:
Description	
Supportanfrage	
<p>uns ist aufgefallen, dass es für Konzernmitglieder DW-Lizenzen gibt und für externe Partner EB-Lizenzen. Damit einher gehen auch zwei verschiedene Artifactorys mit verschiedenartigen ADTF-Paketen. Können die Conan ADTF-Pakete von EB und DW von beiden Parteien verwendet werden oder haben sie jeweils ihren eigenen exklusiven Lizenzierungsmechanismus?</p> <p>Falls inkompatibel: Wie sieht dann ein Zusammenarbeitsmodell mit Internen + Externen Mitarbeitern aus, wenn zwei inkompatible ADTF-Pakete verwendet werden müssen? Ein ADTF-Plugin kann ja nur eine der beiden ADTF-Pakete als Abhängigkeit haben. Ist es möglich, EIN ADTF Paket mit beiden Lizenzmechanismen bereitzustellen? Oder wie soll das in Zukunft gelöst werden?</p>	
Lösung	
License/Package-Mismatch	
<ul style="list-style-type: none">• Kurzfristige Lösung: Falk wird Conan Recipe so anpassen, dass zwischen Lizenzen/Paketen unterschieden werden kann<ul style="list-style-type: none">◦ Kontakt EB: Weißert, Thorsten <Thorsten.Weissert@elektrobit.com>• Mittelfristige Lösung: Falk wird auf EB zugehen, dass deren Licenseser dll/so so angepasst wird, auch die EB Lizenz zu unterstützen (Option 1) bzw. die EB-Lizenz mit dem DW-Paket kompatibel ist (Option 2, bevorzugt)• Langfristige Lösung: Als Thema im Kufo diskutieren, da in absehbarer sicherlich alle Mitglieder und somit deren Dienstleister davon betroffen sein werden• Parallel: DW prüft, ob der Licenser als Extra Paket/Abhängigkeit ausgelagert werden kann	
Lizenz im (Post)Build benötigt	
<ul style="list-style-type: none">• Rechtlich benötigt man ein Lizenz zum Build• Der Plugin Description Generator ist frei von Lizenz, lädt aber u.U. lizenzierte Plugins (je nach Dependency)• Die Generierung ließe sich umgehen (händisch, eigene Mechanismus), wie raten aber zum CMake Post Build Step um sicherzustellen, dass alles funktioniert (auch dann im CE)• Unabhängig davon verstößt man wie erwähnt gegen Lizenzvereinbarungen• Deshalb müssen auch sämtliche Build Maschinen lizenziert werden (technisch, rechtlich)• Der Use Case des Buildvorgangs in der Cloud/Docker ist bei DW am Schirm und wird im neuen Lizenzmechanismus beachtet	
Mit ADTF können nicht alle conan Features genutzt werden	
<ul style="list-style-type: none">• Fokus von ADTF: Build mit CMake• Conan derzeit nur Mittel zum Zweck: Dependency Handling, Build Umgebung, Deployment/Artifactory-Anbindung• Wir stellen es trotzdem bereit und decken so indirekt den geforderten Bedarf, der aber bisher noch nicht ausreichend ist• Deshalb bringt Pierre das Thema conan (grundsätzlich) ins Kufo• Parallel: Falk soll bitte Wünsche und Uses aufschreiben, gerne auch mit einem Bsp. Recipe, dann lässt sich das gemeinsam erarbeiten<ul style="list-style-type: none">◦ #9404	
Related issues:	
Related to Public Support - Support Request #9404: Conan ADTF and SDK package...	Closed

History

#1 - 2020-04-03 15:17 - hidden

- Status changed from New to In Progress

- Topic set to ADTF::License

#3 - 2020-04-06 08:22 - hidden

- Status changed from In Progress to Customer Feedback Required

Hallo Falk,

es gibt keine unterschiedlichen Pakete, diese sind identisch.

Natürlich können wir nicht garantieren, dass AUDI die Pakete ggf. in Zukunft anpasst oder bereitstellt.

Im Zweifel immer die von uns verwenden.

So oder so... der Schutz erfolgt immer anhand der Lizenz und nicht der Binaries.

Eine Kunde mit Nicht-Konzern Lizenz sollte jederzeit auch die Konzern Lieferung verwenden können, sollte es hier Unterschiede geben, muss das über Lizenzflags geregelt sein

#4 - 2020-04-06 08:23 - hidden

Nachtrag zur Vollständigkeit:

Es gibt nicht nur EB Lizenzen sondern mehr Vertriebspartner -> <https://www.digitalwerk.net/adtf/vertriebspartner/>

Alle haben einen eigenen Lizenzmechanismus, der aber mit der ADTF License Lib kompatibel sein muss, auch wenn künftig ein neuer Lizenzmechanismus entsteht.

#5 - 2020-04-07 10:16 - hidden

Danke für die Klarstellung. Ich habe es so weitergegeben und warte auf Feedback, ob die DW-Pakete mit EB-Lizenzen funktionieren. Werde das Ticket dann aktualisieren, sobald ich Neues weiß.

#6 - 2020-04-09 12:40 - hidden

Ich lass mal noch bis nach Ostern offen, aber hier ist denk ich alles gesagt.

Das muss gehen, im Zweifel muss eine Licenser library (die von EB) in unserer Lieferung unsere ersetzen, mehr kann nicht schief gehen.

@Michael: Kann nach dem Due Date geschlossen werden

#7 - 2020-04-09 16:45 - hidden

Verstehe ich das richtig, dass in das DW-ADTF-Paket manuell bei jedem EB-Lizenz-Nutzer (also all unseren externen Partnern) noch eine licenser.dll reingepatcht werden muss ? Oder wird das DW-Paket dahingehend angepasst, dass die EB-DLL standardmäßig drin ist und somit von beiden Parteien (int und ext) genutzt werden kann?

#8 - 2020-04-13 21:36 - hidden

Hallo Falk,

Im DW Paket ist natürlich auch die DW Licenser DLL/SO. Wir geben das Paket 1:1 an unsere Distributoren, so wie es für die ganze Welt verfügbar ist. Wenn ein Distributor eine eigene Logik und eine abweichende Lizenzierung mit eigener Licenser DLL/SO verwenden, müssen sie diesen auch reinpacken und austauschen, was sie soweit ich informiert bin auch tun. Der grundsätzliche Lizenzmechanismus (ADTF License Lib API) im Backend ist aber gleich, nur die Licenser DLL kann spezifisch sein und eine gewisse Logik bereitstellen. Es müssen allerdings die gleichen Flags natürlich implementiert werden.

Ich meinte zur Not müsst ihr einmal ein Pakte lokal patchen, man zieht das ADTF ja nur einmalig.

Sofern EB den Lizenzlibrary austauscht, gibt's keine generische Lösung

Von uns kommt die generische Lösung, die spezifische vom Distributor, d.h. hier muss die Abhilfe geschaffen werden. Rechtlich gibt es keine Probleme, dass ihr zusammenarbeitet.

Meiner Meinung nach müsstet ihr im recipe bei der Dependency lediglich den DW und den Distributor Channel unterscheiden, damit könnt ihr beide Abhängigkeiten nutzen.

Vielleicht ist es aber noch einfacher und die EB Licenser DLL/SO und deren Mechanismus ist kompatibel. Das sollte aber der EB Support beantworten bzw das lässt sich ja relativ schnell herausfinden, wenn eure Partner ein DW Paket mit einer EB Lizenz benutzen und zb den CE starten.

#9 - 2020-04-14 11:33 - hidden

Vielleicht ist es aber noch einfacher und die EB Licenser DLL/SO und deren Mechanismus ist kompatibel. Das sollte aber der EB Support beantworten bzw das lässt sich ja relativ schnell herausfinden, wenn eure Partner ein DW Paket mit einer EB Lizenz benutzen und zb den CE starten."

- DW und EB-Lizenzen sind nicht kompatibel, die DLL muss überschrieben werden.
- EB-Support sagt, es ist **weder vorgesehen noch möglich**, die DW-Pakete zu verwenden

Ich weigere mich aktuell, die zur Verfügung gestellten Pakete zu "patchen". Wo sollen die gehostet werden, wer übernimmt den Support und die Aktualisierungen? Zumal das gepatchte Paket dann auch nur DW **oder** EB unterstützt. Oder man würde es on-the-fly im conan-install beim jeweiligen Nutzer zurechtbiegen, was aber auch wieder zu lokalen Änderungen am Paket führt.

Alternativ im Conan-Rezept des Plugins die zwei Paket-Dependencies zu unterschieden wird nicht sauber gehen, da je nach verwendeten ADTF-Paket eine andere Package-ID generiert wird und diese dann inkompatibel zueinander sind. Mal von der "Qualität der Lösung" abgesehen (if adtf_lic == "EB": ...)

Unser Anspruch ist, dass alle Projektpartner mit dem selben **von DW bereitgestellten** Conan-Paketen arbeiten. Ohne Hacks in den Dependencies oder den Paketen. Wir arbeiten mit diversen Partnern gemeinsam an den Plugins und Configs, die möglichen Fehlerursachen durch verschieden gepatchte Pakete will ich gern von vornherein ausschließen.

#10 - 2020-04-14 11:42 - hidden

- Status changed from Customer Feedback Required to In Progress

#11 - 2020-04-14 11:52 - hidden

ist denn bekannt, ob die Pakete der anderen Distributoren die DW-License-DLL verwenden? Es wäre ja auch eine Option, dass sich die externen Partner von EB zu verabschieden und bei einem anderen Dist die Lizenzen kaufen.

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

- Status changed from In Progress to Customer Feedback Required

Hallo Falk,

ist denn bekannt, ob die Pakete der anderen Distributoren die DW-License-DLL verwenden? Es wäre ja auch eine Option, dass sich die externen Partner von EB zu verabschieden und bei einem anderen Dist die Lizenzen kaufen.

Soweit mir bekannt, wird bei den anderen aktuell nichts ausgetauscht. Zu einen Wechsel kann und werde ich mich seitens Digitalwerk nicht äußern im Sinne unserer Partnerschaften und Gleichberechtigungen am Markt, das sollte dann der jeweilige Partner beantworten, siehe <https://www.digitalwerk.net/adtf/vertriebspartner/>

Man sollte aber Bedenken, dass damit erneute Lizenzkosten (die von EB hat man ja schon erworben) einhergehen. Für künftige Lizenzen kann man das unabhängig von diesen Use Case ohnehin überdenken und anfragen.

Ich weigere mich aktuell, die zur Verfügung gestellten Pakete zu "patchen". Wo sollen die gehostet werden, wer übernimmt den Support und die Aktualisierungen? Zumal das gepatchte Paket dann auch nur DW oder EB unterstützt. Oder man würde es on-the-fly im conan-install beim jeweiligen Nutzer zurechtbiegen, was aber auch wieder zu lokalen Änderungen am Paket führt.

Ich meinte mit dem patchen nur lokal, das zieht man sich ja nur einmal im cache. Das wäre auch nur die Übergangslösung für diesen Use Case, der wie du ja EB zitierst bisher keine Anforderung war und ist. Ein eigenes Paket hosten und pflegen und supporten würde ich vermeiden, ist aber nichts anderes, als ihr im 2er mit den Sandboxen gemacht habt.

Alternativ im Conan-Rezept des Plugins die zwei Paket-Dependencies zu unterschieden wird nicht sauber gehen, da je nach verwendeten ADTF-Paket eine andere Package-ID generiert wird und diese dann inkompatibel zueinander sind. Mal von der "Qualität der Lösung" abgesehen (if adtf_lic == "EB": ...)

Also ehrlich gesagt fallen mir hier viel schlimmere und unsaubere Dinge ein, die im Konzern laufen... und damit meine ich nicht nur die Sandboxen :-)

Aber siehst du das wirklich so ?

Ich meine man schreibt ein recipe, in der eine Fallunterscheidung ist, dann zieht man sich entweder z.B. ADTF/3.7.0@dw/stable oder ADTF/3.7.0@eb/stable.

Ich finde das alles andere als unsauber und dann können alle mit dem gleichen recipe Stand arbeiten.

Ich würde sogar behaupten, so läuft die Welt und Dependency Management, je nach Anspruch, Verfügbarkeit und Ausprägung.

Alternativ könnte man noch ein Basis recipe schaffen und zwei recipe Spezialisierungen (EB/DW), welche die ADTF dependency definieren. Hier wäre es klar getrennt und die Pflege ausschließlich im Basis recipe, die Spezialisierung übergibt nur den Channel.

Wenn EB auch einen Channel anbieten würde, der dw heißt, musst du imho sogar gar nichts anpassen und eure Partner binden nur das EB Artifactory ein (gibts das überhaupt?).

Mit der Package ID hat das imho nichts zu tun, man switched nur den verfügbaren Channel.

Unser Anspruch ist, dass alle Projektpartner mit dem selben von DW bereitgestellten Conan-Paketen arbeiten. Ohne Hacks in den Dependencies oder den Paketen. Wir arbeiten mit diversen Partnern gemeinsam an den Plugins und Configs, die möglichen Fehlerursachen durch verschieden gepatchte Pakete will ich gern von vornherein ausschließen.

Das wird wie gesagt technisch nicht möglich sein, wenn man EB die Kompatibilität nicht aufzwingt (aufzwingen kann). Vertraglich weiß ich überhaupt nicht was das bedeutet, ist aber auch nicht in meinen Verantwortungsbereich.

Das wäre eine völlig neue Anforderung und muss über deinen Kundenforumsvertreter auch ins Kundenforum gebracht werden. Hier müsste nicht nur der Paket und Lizenz Support eingespeist werden, sondern auch grundsätzlich conan Support, das bekommt ihr aktuell nur indirekt, weil wir mit conan entwickeln, das Kundenforum fordert eine Archive Delivery und dann stellen wir das quasi 1:1 bereit, um euch indirekt zu bedienen.

Eigentlich wäre das Aufgabe der AEV, für AUDI bzw. VW alles bereitzustellen, da hier der conan Prozess definiert ist. Aber warum doppelte Arbeit denken wir uns...

Künftig werden wir es so anpassen, dass es auch ein SDK only für den build gibt und das eher Laufzeit-relevante "all-in-package" mit mixed build type inklusive tools und co (bishere Lieferung) als reines Archiv. Das aber auch alles abseits sämtlicher Kundenforumsanforderungen/-budget, nur um dich mal diesbezüglich abzuholen. Deshalb wäre es hilfreich, dass ihr das via Konzernvertreter einspeist, um hierfür Ressourcen zu bekommen, es anständig und zielführend hochzuziehen.

Eine gleichzeitige Verwendung des selben Paketes wird heute und künftig nur gehen: Gleicher/kompatibler Lizenzmechanismus, gleiches/unverändertes Paket

Das ist aber ein reines Distributorthema.

Ich würde es gerne mit in die Überlegungen des kommenden neuen Lizenzmechanismus aufnehmen, brauche dazu aber eine offiziell eingespeiste Anforderung.

Hilft euch aber derzeit auch nicht weiter.

Aktuell sehe ich nur folgende Möglichkeiten:

- recipes so anpassen, dass sie die kompatiblen Dependencies anziehen (würde ich empfehlen)
- Dienstleister arbeitet mit kompatiblen Paketen/Lizenzen
- Ihr gestaltet eure Projekte (finanziell) so, dass die Dienstleister die Lizenz vom Konzernpool als Beistellung bekommen und könnt beide am DW Paket arbeiten

Für mehr sehe ich bei uns die Hände gebunden, würde aber gerne zu all euren Anforderungen diesbezüglich mal eine Telko machen, um das ganze auch vertraglich/technisch zu betrachten, zusammen mit Tobias Schmid.

Wie siehst du das Falk ?

@Tobi: Anregungen ?

#13 - 2020-04-20 21:48 - hidden

Zeit für Feedback nochmal verlängert

#15 - 2020-04-23 16:52 - hidden

Hi Florian,

erstmal Danke für deine detaillierte Antwort.

Wir haben uns bei den anderen Distributoren umgehört und diese nutzen den DW-Lizenzmechanismus. Die Entscheidung von EB wegzugehen liegt dann halt bei uns. Da es keine Anforderung zur Kompatibilität der Lizenzmechanismen gibt, kann sich das aber eben auch jederzeit ändern.

Wegen Patchen im Cache: Wenn man zuverlässig mit Conan arbeiten will, sollte der Cache ReadOnly sein, um eben genau lokale Unterschiede bei den Entwicklern auszuschließen. Das halte ich also für keine gute Idee. Außerdem wissen wir ja, wenn man erstmal mit sowas anfängt, wird dann lokal mit der Zeit hier und da noch mehr 'gepatcht', weil der Entwickler denkt, das löst irgend ein Problem. Und nur weil das Durchschnittslevel der Script-/Tool-Qualität "nicht perfekt" ist, würde ich mich ungern von vorn herein auf das selbe Niveau begeben :)

Problem bei der Fallunterscheidung im conanfile ist doch, dass ein Filter, der mit requires(ADTF@dw) arbeitet, eine andere package-id generiert als mit requires(ADTF@eb). Im Artifactory müssten dann immer beide Varianten von einem Jenkins gebaut werden. Alternativ könnte man das ADTF aus der package-id werfen oder zu einem build_requires() machen, sodass es nicht bei der package-id berücksichtigt wird. Beides sind meiner Meinung nach unsaubere Hacks (funktionieren aber zumindest).

Wenn es ein SDK-Paket ("header&libs") gäbe, was nur zum Bauen gebraucht wird, könnte man zumindest an der Stelle auf das doppelte Bauen oder Patchen der package-id verzichten. Im Moment braucht man ja sogar eine Lizenz, um die plugindescription zu generieren (der nächste Dorn in meinem Auge).

Wer ist denn der Kanal, um das Lizenzmechanismus-Thema und Lizenz-zum-Bauen-Thema einzuspeisen ... Alex?

#16 - 2020-04-23 17:35 - hidden

Hi Falk,

Wegen Patchen im Cache: Wenn man zuverlässig mit Conan arbeiten will, sollte der Cache ReadOnly sein, um eben genau lokale Unterschiede bei den Entwicklern auszuschließen. Das halte ich also für keine gute Idee. Außerdem wissen wir ja, wenn man erstmal mit sowas anfängt, wird dann lokal mit der Zeit hier und da noch mehr 'gepatcht', weil der Entwickler denkt, das löst irgend ein Problem. Und nur weil das Durchschnittslevel der Script-/Tool-Qualität "nicht perfekt" ist, würde ich mich ungern von vorn herein auf das selbe Niveau begeben

Ja, bin ich bei dir, war wie gesagt eine absolute Notlösung aus dem ersten Schuss...

Problem bei der Fallunterscheidung im conanfile ist doch, dass ein Filter, der mit requires(ADTF@dw) arbeitet, eine andere package-id generiert als mit requires(ADTF@eb).

OK das wusst ich nicht

Im Artifactory müssten dann immer beide Varianten von einem Jenkins gebaut werden.

Hol mich mal ab... warum ist das ein Problem ?

In meiner Welt ist doch egal wer den Source Code baut bzw. das binary liefert.
Du definierst eine Abhängigkeit zu einem Produkt, Version und Channel ?

Ich habe den Use Case so verstanden:

Ihr entwickelt gemeinsam Filter oder tauscht binaries aus.

Das ist doch so oder so problemlos möglich ?

Sowohl der Build (mit Schalter im Recipe) als auch die Runtime (Session ohnehin unabhängig).

Vielleicht stehe ich auch auf der Leitung aber ich sehe nirgends einen Hash oder Package ID...

Alternativ könnte man das ADTF aus der package-id werfen oder zu einem build_requires() machen, sodass es nicht bei der package-id berücksichtigt wird. Beides sind meiner Meinung nach unsaubere Hacks (funktionieren aber zumindest).

Gut, wir arbeiten bei uns eigentlich ausschließlich mit build_requirements, das aber nur zur Vollständigkeit.

Demnach ist es bei uns egal, wo was liegt, definiert wird nur ADTF/x.y.z@bla/stable.

Nun kann ich aus X-bla-Quellen das Produkt beziehen, der eine definiert unseren Channel, der andere zb ein bintray, der nächste die CIP...

Wenn es ein SDK-Paket ("header&libs") gäbe, was nur zum Bauen gebraucht wird, könnte man zumindest an der Stelle auf das doppelte Bauen oder Patchen der package-id verzichten.

Versteh ich leider nicht... warum musst du doppelt bauen ?

Im Moment braucht man ja sogar eine Lizenz, um die plugindescription zu generieren (der nächste Dorn in meinem Auge).

Das ist so korrekt ! Der Plugin Description Generator fährt eine Runtime hoch und prüft die Dependencies.

Schau mal in deine Lizenz, natürlich musst du auf dem Build System eine ADTF Developer Lizenz haben.

Das lässt sich auch wunderbar abbilden.

Im Prinzip haben wir nun implizit endlich einen Schutz dafür :-)

Wer ist denn der Kanal, um das Lizenzmechanismus-Thema und Lizenz-zum-Bauen-Thema einzuspeisen ... Alex?

In allen Belangen werdet ihr von Pierre vertreten, ich schlage aber dennoch einen gemeinsamen Termin vor, unabhängig davon, dass er es dann "offiziell einspeist".

Mir sind noch zu viele Fragezeichen und wir brauchen eine gemeinsame Basis wie du siehst, ich sehe das Problem wie versucht zu schildern nicht an der Lösung bzw. warum sie für dich eher unsauber ist.

Technisch werden wir das Lizenz-implimentierungsthema (Mechanismus im Backend ist ja der gleiche) nicht lösen können denke ich, d.h. man müsste das vertraglich ändern.

Ggf. muss man weiter beim Thema open source anstoßen, dann könnte das ja an der ein oder anderen Stelle wegfallen.

Ändert aber nichts daran, dass auch ein SDK only Package einen Schutz braucht.

Wir haben das auch schon soweit fertig und müssen das mal bewerten.

Problem ist man braucht trotzdem die plugins (wegen pdgen) aber auch qt (für ui plugins)...

Aber ohne pdb und tools siehts schon mal besser aus, in der 3.7 wäre das (RelWithDebInfo/Debug):

- Windows -> 350/420 MB
- Linux (Desktop) -> 750/630 MB
- Linux (ARMv8) -> 470/375 MB

Man könnte dann noch ein ADTF_SDK, ADTF_UISDK Package machen aber das sind dann ein paar MB hin oder her...

Werden wir aber bei der neuen Lizenzthematik bzgl. Qt bis zum nächsten Kundenforum aufbereiten.

#17 - 2020-04-27 18:11 - hidden

- Support Level changed from 2nd Level to 3rd Level

Termin zur Durchsprache erstellt

#18 - 2020-04-29 16:12 - hidden

Absprache am 29.04.2020 (Falk, Ralf, Tobi, Flo):

License/Package-Mismatch

- Kurzfristige Lösung: Falk wird Conan Recipe so anpassen, dass zwischen Lizenzen/Paketen unterschieden werden kann
 - Kontakt EB: Weißert, Thorsten <Thorsten.Weissert@elektrobit.com>
- Mittelfristige Lösung: Falk wird auf EB zugehen, dass deren Licenseser dll/so so angepasst wird, auch die EB Lizenz zu unterstützen (Option 1) bzw. die EB-Lizenz mit dem DW-Paket kompatibel ist (Option 2, bevorzugt)
- Langfristige Lösung: Als Thema im Kufo diskutieren, da in absehbarer sicherlich alle Mitglieder und somit deren Dienstleister davon betroffen sein werden
- Parallel: DW prüft, ob der Licenseser als Extra Paket/Abhängigkeit ausgelagert werden kann

Lizenz im (Post)Build benötigt

- Rechtlich benötigt man ein Lizenz zum Build
- Der Plugin Description Generator ist frei von Lizenz, lädt aber u.U. lizenzierte Plugins (je nach Dependency)
- Die Generierung ließe sich umgehen (händisch, eigene Mechanismus), wie raten aber zum CMake Post Build Step um sicherzustellen, dass alles funktioniert (auch dann im CE)
- Unabhängig davon verstößt man wie erwähnt gegen Lizenzvereinbarungen
- Deshalb müssen auch sämtliche Build Maschinen lizenziert werden (technisch, rechtlich)
- Der Use Case des Buildvorgangs in der Cloud/Docker ist bei DW am Schirm und wird im neuen Lizenzmechanismus beachtet

Mit ADTF können nicht alle conan Features genutzt werden

- Fokus von ADTF: Build mit CMake
- Conan derzeit nur Mittel zum Zweck: Dependency Handling, Build Umgebung, Deployment/Artifactory-Anbindung
- Wir stellen es trotzdem bereit und decken so indirekt den geforderten Bedarf, der aber bisher noch nicht ausreichend ist
- Deshalb bringt Pierre das Thema conan (grundsätzlich) ins Kufo
- Parallel: Falk soll bitte Wünsche und Uses aufschreiben, gerne auch mit einem Bsp. Recipe, dann lässt sich das gemeinsam erarbeiten

#19 - 2020-04-29 21:40 - hidden

zum letzten Punkt habe ich die [#9404](#) noch einmal ergänzt.

#20 - 2020-05-06 08:21 - hidden

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

#21 - 2020-05-06 08:22 - hidden

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

#22 - 2020-05-06 08:22 - hidden

- Description updated

#26 - 2020-07-07 12:49 - hidden

- Status changed from To Be Closed to Closed