

Programmierrichtlinien in Tcl-Skripten bei Kundenanpassungen

Abas APS 2024.Q3.1

VERSION: APS2201R8N03

STAND: 29.07.2025

1 ALLGEMEINE RICHTLINIEN

- Tcl kennt keine Datentypen, daher ist es ratsam Arrays und Listen schon am Variablennamen zu erkennen. Arrays beginnen mit Großbuchstaben und enden auf **Array** (Bsp. KoppelMgrArray). Arrays, die nicht global sind, sondern nur innerhalb einer Prozedur genutzt werden, beginnen mit **Tmp** (Bsp. TmpKoppelMgrArray). Listen beginnen mit Kleinbuchstaben und enden auf **List** (Bsp. setupList). Eine derartige Kennzeichnung existiert nicht für primitive Datentypen (Integer/String/usw.). Variablen von primitiven Datentypen beginnen mit Kleinbuchstaben.
- Konstanten werden komplett großgeschrieben (Bsp. CUSTOMER).
- Die öffnenden Klammern von Prozeduren, Schleifen und if-Anweisungen wird am Zeilenende platziert, die schließende Klammer in einer separaten Zeile:

```
foreach j [job used] {  
    ...  
}
```

- Sicherungen von Skripten dürfen nicht im Skriptverzeichnis gespeichert werden, sondern in einem separaten Unterverzeichnis.

2 RICHTLINIEN FÜR KUNDENANPASSUNGEN

- Für jeden Kunden mit Kundenanpassung wird ein Kundenkürzel in Großbuchstaben vergeben (Bsp. FTL).
- Skriptanpassungen erfolgen immer in zusätzlichen Skripten, welche stets mit dem Kundenkürzel und Unterstrich beginnen (Bsp. FTL_simImpTech.tcl).
- Müssen für die Kundenanpassung zusätzliche Daten eingelesen werden, dann erfolgt dies ausschließlich in dem Skript **<Kundenkürzel>_simImpCustomizingData.tcl** (Bsp. FTL_simImpCustomizingData.tcl) mit der Prozedur **createCustomizingData<Kundenkürzel>** (Bsp. createCustomizingDataFTL). Hierin ist kurz zu dokumentieren, welche Daten zusätzlich eingelesen werden und wofür sie verwendet werden. Die ausführliche Dokumentation erfolgt in dem Skript, welches die Daten verarbeitet.
- Kundenindividualisierte Schnittstellendateien (nur Professional) erhalten ein zusätzliches **.C.<Kundenkürzel>** (Bsp. EDPEXPORTAGSTAMM.C.FTL.20200515080946)

- Für fundamentale Prozeduren existieren Individualisierungsmechanismen in den Standardskripten, sodass individualisierte Prozeduren der Form **<Prozedurname><Kundenkürzel>** (Bsp. **agplanenFTL**) statt ihrer Standardvarianten abgearbeitet werden. Dabei ist auf folgende Konstrukte zu achten:

```
if { $CUSTOMER != "" && [info proc agplanen$CUSTOMER] != "" } {  
    set planen [agplanen$CUSTOMER $element $idResList]  
} else {  
    set planen [agplanen $element $idResList]  
}
```

Soll die Prozedur **agplanen** nun im Rahmen einer Kundenanpassung überschrieben werden, dann muss der Programmierer lediglich eine Prozedur **agplanenFTL** schreiben und diese in einem Skript **FTL_simImpTech.tcl** im Skriptverzeichnis hinterlegen (**FTL_simImpTech.tcl** deshalb, weil die Standardprozedur **agplanen** im Standardskript **simImpTech.tcl** zu finden ist).

In seltenen Fällen kann es vorkommen, dass aus design-technischen Gründen neue Einstiegspunkte für Individualisierungen in den Standardskripten geschaffen werden müssen/sollten. Dies sollte nur in Absprache mit den Programmierverantwortlichen der Forterro Deutschland Abas GmbH erfolgen.

- Bei Hilfsprozeduren, die in Kundenangepassten Prozeduren benötigt werden, wird das Kundenkürzel vorangestellt statt hinten angehängen.

```
proc createCustomizingDataFTL {DataDir ppid} {  
    #Anpassung FTL  
    #-> Unterlieferung/Überlieferung aus dem Kundenstamm  
    FTLimportKdStamm $DataDir $ppid  
}  
...  
proc FTLimportKdStamm {DataDir ppid} {  
    #Einlesen der Daten und speichern im globalen Array  
    ...  
}
```

Durch das Voranstellen des Kundenkürzels (**FTLimportKdStamm**) wird verhindert, dass diese Prozedur fälschlicherweise durch die Einstiegspunkte in den Standardskripten aufgerufen wird.

- Kundenangepasste Variablen an den Modellierungsobjekten im Simulationsmodell beginnen mit dem Kundenkürzel (Bsp. **machinex :: set FTLsetupMaxCount 59**).
- Kundenanpassungen sind ausreichend im Code zu dokumentieren. Werden dabei Standardprozeduren verwendet und nur teilweise abgeändert, so sind die geänderten Blöcke zu markieren.
- Im Upgradefall sind nur kundenindividualisierte Skripte zu ändern.