

Programming guidelines for customizing Tcl scripts

Abas APS 2024.Q3

VERSION: APS2200R8N02

DATE: 10.04.2025

1 GENERAL GUIDELINES

- Tcl does not recognize data types, so we advise making arrays and lists recognizable by their variable names. Arrays begin with capital letters, and end with **Array** (e.g. KoppelMgrArray). Arrays that are not global, but only used within a procedure, begin with **Tmp** (e.g. TmpKoppelMgrArray). Lists begin with lower case letters and end with **List** (e.g. setupList). This kind of labeling does not exist for primitive data types (integer/string/...) Primitive data type variables begin with lower case letters.
- Constants are fully capitalized (e.g. CUSTOMER).
- The opening parenthesis of procedures, loops and if-statements is placed at the end of the row, the closing parenthesis in a separate row:

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

- Script backups may not be saved in the script directory, but in a separate subdirectory.

2 CUSTOMIZING GUIDELINES

- Each customer using customizations has a customer abbreviation in capital letters (e.g. FTL).
- Scripts may only be customized using separate scripts that begin with the customer code and an underscore (e.g. FTL_simImpTech.tcl).
- If the script loads additional data for the customization, it should only be performed in this script:
<Customer code>_simImpCustomizingData.tcl (e.g. FTL_simImpCustomizingData.tcl), with this procedure:
createCustomizingData<customer code> (e.g. createCustomizingDataFTL).
This script should contain a brief explanation of what data is loaded additionally and what it is used for. The script that processes the data should contain more detailed documentation.
- Customized interface files (only professional) must have an additional suffix: **.C.<customer code>** (e.g. EDPEXPORTAGSTAMM.C.FTL.20200515080946).
- Standard scripts for fundamental procedures have customizing mechanisms, so that custom procedures named as **<procedurename><customer code>** (e.g. agplanenFTL) are processed instead of the standard procedures. Ensure that the following constructs are observed:

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

If you overwrite the **agplanen** script as part of the customization, then you only need to write an **agplanenFTL** procedure, for example, and store it in an **FTL_simImpTech.tcl** script in the following script directory: **TFL_simImpTech.tcl**. This is because the standard procedure **agplanen** is found in the standard script: **simImpTech.tcl**.

In rare cases new points of entry may be required for design/technical reasons for customizations to standard scripts. This should only be done after consultation with the programming managers at Forterro Deutschland Abas GmbH.

- Auxiliary procedures required by custom procedures should have the customer code as a prefix instead of a suffix.

```
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  
... }
```

Using the customer code as a prefix (e.g. **FTLimportKdStamm**) prevents the program from mistakenly calling this procedure by the entry points instead of the standard scripts.

- Custom variables in modeling objects in the simulation model must begin with the customer code (e.g. `machinex :: set FTLsetupMaxCount 59`).
- Customizations must be adequately documented in the code. If you use standard procedures and only partially change them, the modified blocks must be marked.
- In the event of an upgrade, only customized scripts should be changed.