

TKC User Files and Directories

Table of Contents

- 1: Introduction
- 2: Overall Structure of TKC
- 3: Project or User Directory Contents
- 4: User Config File *_TKC_User_Data.cfg*
- 5: Contents of Sub-Dirs. of a TKC Project Dir.
- 6: Contents of the *TKC_Root* Directory
- 7: ADAMS/View Shortcuts and Final Remarks

1: Introduction

In this section, we will explain the overall structure of the program and the relevant files and directories used by TKC.

2: Overall Structure of TKC

This section briefly explains the overall structure of TKC.

- The Acronym TKC stands for *ToolKit Creator*; it is an ADAMS View-based customization toolkit.
- TKC is created and maintained by Sayfield International and is being used by us and some Benelux ADAMS customers for several projects and customizations of ADAMS View.
- TKC models contain parametric components that are either created on-the-fly or from predefined assembly files as will be illustrated later in this manual.
- All TKC functionality and data files are defined in ADAMS/View command language ASCII macros and stored in toolkit directories
- The TKC backbone functionality, as well as each toolkit, is stored in the ADAMS/View binary as a library. This allows storage of toolkits and libraries separately from the main ADAMS binary.

3: Project or User Directory Contents

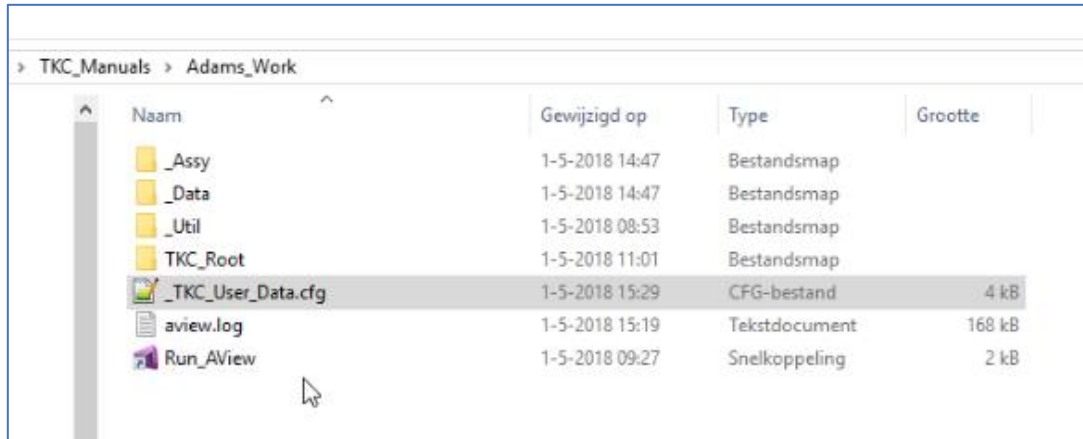
For user projects, TKC is delivered in a self-contained ADAMS project directory. This directory is the typical directory where users will perform their TKC simulation projects. It contains sub-directories to store utilities and data, and stores project related models, and model in- and output files. Some files in the project dir. define the TKC macro functionality. As TKC is a custom ADAMS/View, the file *aview.cmd* is pre-defined in ADAMS/View and loads dedicated objects and tools. The code sub-dir. *TKC_Root* in the project dir. stores the code files of TKC. *TKC_Root* can be moved to a central location to make a *global* installation.

Typical sub-directories you will find in a TKC Project directory are *_Util*, *_Data* and *_Assy*. These are reserved names, and the underscore serves to let these directories *float to the top* so users can find them more easily. These directories typically contain:

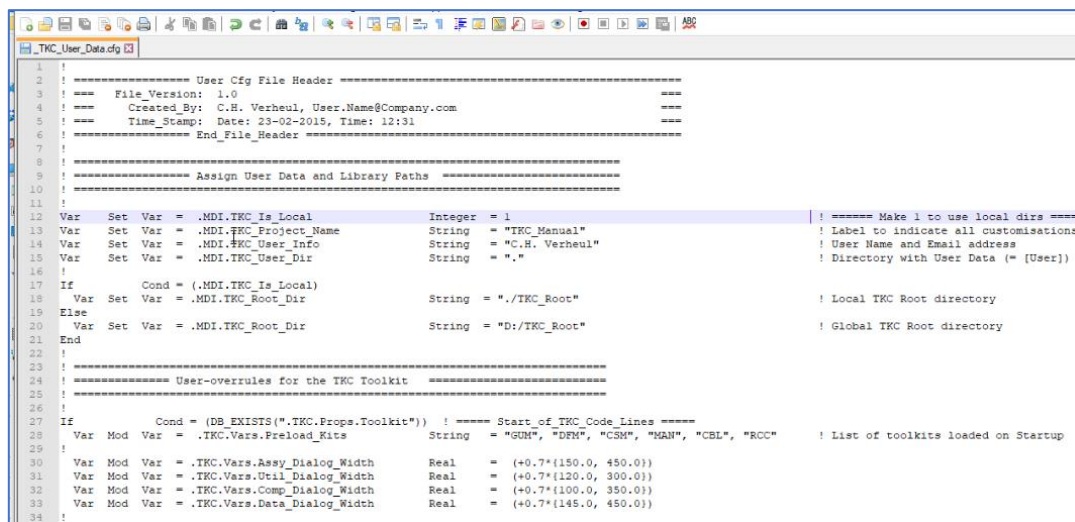
- *_Util*: User utilities and possibly other local TKC toolkits.
- *_Data*: Data for feeding TKC models with required component data.
- *_Assy*: Assembly files to create full TKC models. Assembly files will be explained further.

4: User Config. File `_TKC_User_Data.cfg`

Typical project data or user data is stored in the configuration file called: `_TKC_User_Data.cfg`. This ASCII file is defined in ADAMS command language, and it allows to set several variables in TKC and customize it to project related preferences. The file can either be stored in your local project directory or global in your user home; it typically contains settings to modify options and appearance of TKC to adjust it to your project preferences.



The image above shows a TKC project directory with sub-dirs. and the `_TKC_User_Data.cfg` file.



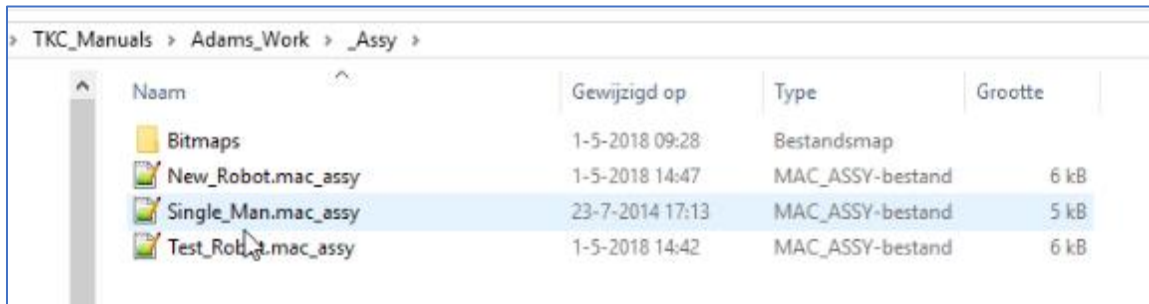
The image above shows the syntax of the configuration file, including comments to enhance readability. A local installation is used as the variable `TKC_Is_Local` is one. Changing it to zero activates TKC to use a global `TKC_Root` dir.

Other values shown are used to set dialog sizes, switches to activate special functionalities such as the `Ribbon_Bar`, the dedicated Toolkits loaded in that `Ribbon_Bar` and switches to set the Modify Callback in this TKC session. As mentioned, this configuration file is stored in the project directory allowing unique configuration files for each project. The configuration file can be changed and reloaded during a TKC session to update configuration values.

Configuration variable `Preload_Kits` defines which Toolkits to load upon starting TKC. Toolkits can also be loaded (and unloaded) in a session using the `TKC Configuration Dialog` or by loading new toolkits objects from assembly files.

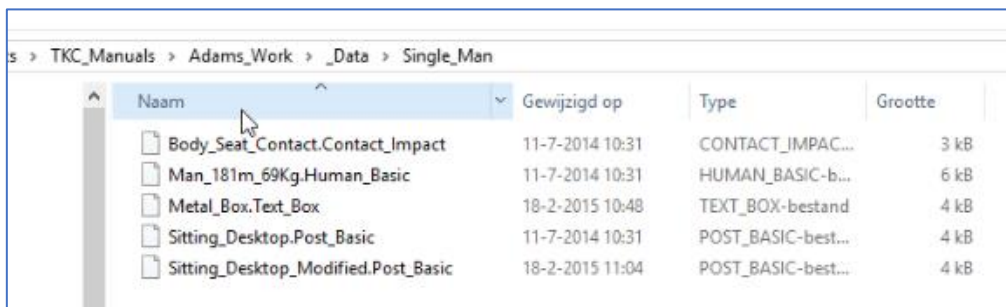
5: Contents of Sub-Dirs. of a TKC Project Dir.

The contents of a TKC project dir. will be briefly illustrated. The *_Assy* directory as mentioned contains readable ASCII Files, defining a full model assembly. The image below shows some assembly files and the *Bitmaps* sub-dir. containing assembly bitmaps.



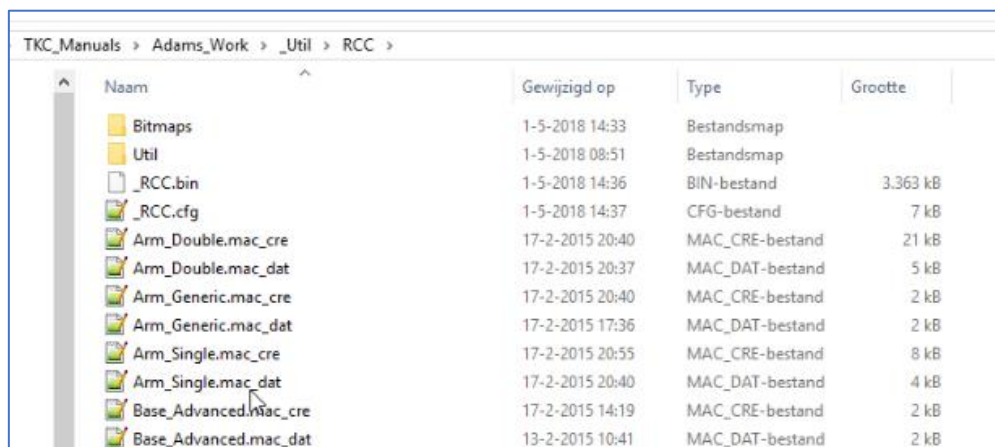
TKC Assembly files can be hand edited using any text editor. The files contain a list of data definitions for model data as well as the model topology the composed of a list of model components. Model data and model components are defined from the execution of assembly file commands. Escape points or groups in Assembly files allow for partial execution of assemblies.

The *_Data* directory contains data files with design variables for model components. In this example, the directory contains sub-directories where a sub-dir. contains data files for one of the assemblies.



The image above shows the data sub-dir. of the *Single_Man* assembly. This is one of the data storage methods that TKC provides. The data files in data sub-dir. *Single_Man* define geometry, stiffness and damping data for the person and for the seat object that the person is seated on in the model.

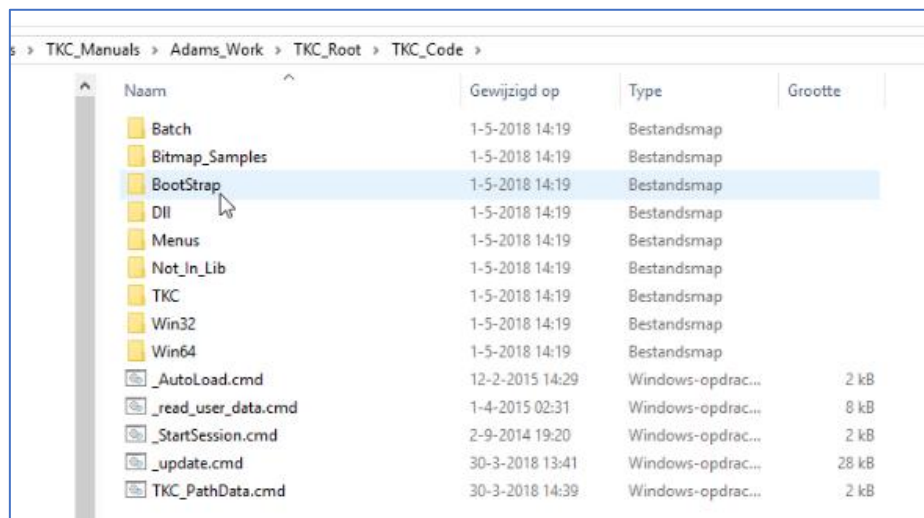
In this example, the directory *_Util* contains the *RCC* User Toolkit with robot components that will be illustrated in this manual.



TKC Toolkit dirs. contain a minimal amount of configuration and macro files. This basic Toolkit contains configuration file *_RCC.cfg* for registration of components and macros with extension **.mac_cre* for the registered component topology and extension **.mac_dat* for registered component data. The *_Util* dir. is also stores project specific utilities. The syntax and definition of utilities is discussed in another section.

6: Contents of the *TKC_Root* Directory

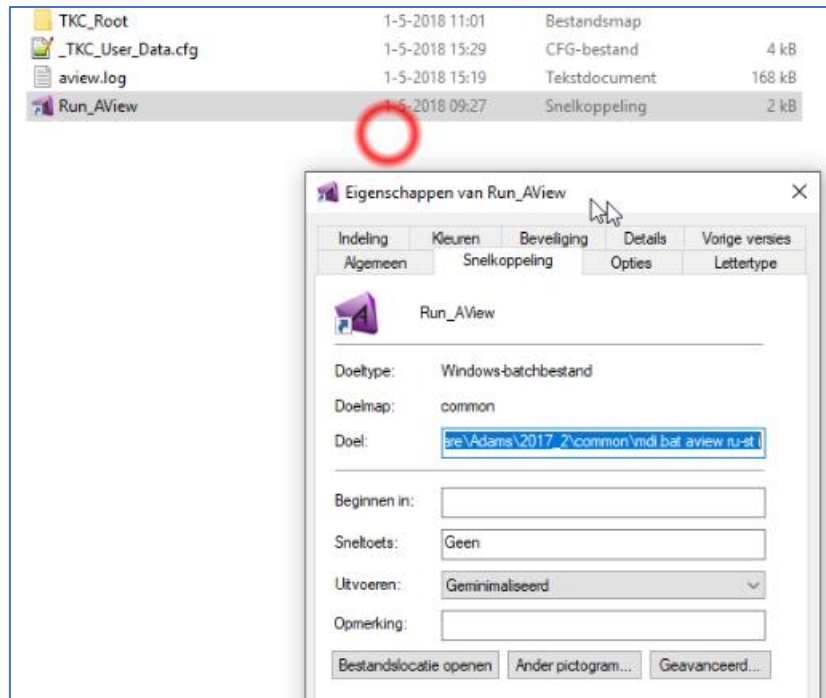
Project directory defined in local mode contains sub-dir. *TKC_Root*. This directory stores the code of the TKC customization and the *Shared* and *Site* pre-defined Toolkit libraries. Here, *TKC_Shared* stores toolkits, data and assemblies for every user, while *TKC_Site* stores objects for a selected group of users. The remaining directory *TKC_Code* contains all TKC Backbone code as illustrated below.



| Naam | Gewijzigd op | Type | Grootte |
|---------------------|-----------------|-------------------|---------|
| Batch | 1-5-2018 14:19 | Bestandsmap | |
| Bitmap_Samples | 1-5-2018 14:19 | Bestandsmap | |
| BootStrap | 1-5-2018 14:19 | Bestandsmap | |
| Dll | 1-5-2018 14:19 | Bestandsmap | |
| Menus | 1-5-2018 14:19 | Bestandsmap | |
| Not_In_Lib | 1-5-2018 14:19 | Bestandsmap | |
| TKC | 1-5-2018 14:19 | Bestandsmap | |
| Win32 | 1-5-2018 14:19 | Bestandsmap | |
| Win64 | 1-5-2018 14:19 | Bestandsmap | |
| _AutoLoad.cmd | 12-2-2015 14:29 | Windows-opdrac... | 2 kB |
| _read_user_data.cmd | 1-4-2015 02:31 | Windows-opdrac... | 8 kB |
| _StartSession.cmd | 2-9-2014 19:20 | Windows-opdrac... | 2 kB |
| _update.cmd | 30-3-2018 13:41 | Windows-opdrac... | 28 kB |
| TKC_PathData.cmd | 30-3-2018 14:39 | Windows-opdrac... | 2 kB |

7: ADAMS/View Shortcut and Final Remarks

The remaining object in a TKC project directory is a Windows shortcut to start ADAMS/View in the TKC ADAMS customization. In this shortcut file, the *start-in* directory is left blank to start ADAMS in the project directory and load the local configuration files mentioned in this section.



This approach is the standard implementation of TKC and is a prerequisite to run TKC properly and use the right project configuration file.

With some experience in ADAMS and Windows, users can modify many of the default settings in TKC. The code directory `TKC_Root/TKC_Code/TKC` contains settings file `_TKC.cfg` with TKC settings. The definition line of many settings can be copied to user configuration files to adjust user preferences. The advised approach to take is to delete all *.bin files defined in TKC, adjust options and reload TKC to evaluate the effects of the changes applied. All settings are commented to document their function.