Release Notes TASverter r2019-07-19_Beta

Introduction

This is a beta release of TASverter that offers:

- a few improvements for the TRASYS reader in handling angle tolerances
- a few bug-fixes for the STEP-TAS geometry viewer
- improvements on TMMverter in processing SINDA/Fluint models, for instance handling LOADQ statements internally, DIV and DPV handling

on top of the previous release offering:

- some minor bug-fixes to the existing TRASYS reader and writer
- a new packaging system allowing some customisation via environment variables
- a STEP-TAS geometry viewer
- TMMverter: a new converter module for TMM model data between ESATAN-TMS and SINDA/Fluint formats.

Note that no changes for any of the other original reader/writer modules in TASverter were made. Therefore these reader/writers are not able to handle the current newer syntax. Users are expected to use their native tool to convert to STEP-TAS format as much as possible and then perhaps use TASverter to convert the STEP-TAS format data to the format they want.

We kindly ask you to provide your feedback on this release to tasverter@thermal.esa.int

TRASYS

The bugfixes to TASverter primarily consider the processing of TRASYS model data. A number of syntax dialects often encountered are taken into account and an output format error has been corrected.

Packaging

This beta release of TASverter is provided as a single executable in a version for Linux and a version for Windows. For each platform this single executable contains the command line version of TASverter as well as the GUI version. No additional setup is needed.
On Linux the command line version is the default whereas Windows defaults to the GUI version. The user can easily change this behaviour by using `--gui` or `--nogui` as the first parameter to the tool.

To better inform users of updates or specific issues the tool can access a page on our system and display the content to the user. By default the tool quickly takes a peak to see if there is any news and if so informs the user, every time it is used. The user can prevent this behaviour by setting an environment variable:

```
TASVERTER_FLAGS=NO_NEWS
```

In that case the user can check manually using `--news` as the first parameter to the tool. In the default setting the user gets a one line message if there is any news and needs to run the tool with the `--news` parameter to read it.

This same environment variable can also be used to set the default GUI behaviour:

```
TASVERTER_FLAGS=NO_GUI
```

if the tool should start as command line version or

```
TASVERTER_FLAGS=GUI
```

if the GUI version is preferred by default. The user can still use the --`gui` or --`nogui` parameter to switch the behaviour. Flags can also be combined by separating them with a comma, for instance

```
TASVERTER_FLAGS=GUI,NO_NEWS
```

will instruct the tool to start as GUI version and not check for news. Note that the order of the flags does not matter except that later flags cancel previous counter parts. For instance

```
TASVERTER_FLAGS=GUI,NO_NEWS,NO_GUI
```

will instruct the tool to start as command line version and not check for news. Here the NO_GUI flag cancels the GUI flag.

System-wide default `TASVERTER_FLAGS` can be set by the administrator and overruled by the user. On linux the user overrules such defaults by using:

```
TASVERTER_FLAGS=$TASVERTER_FLAGS,my-tasverter-flags
```

where `my-tasverter-flags` represents the combination of flags desired by the user. Likewise a Windows user would use:

```
TASVERTER_FLAGS=%TASVERTER_FLAGS%,my-tasverter-flags
```

STEP-TAS viewer

A geometry viewer for STEP-TAS model files is provided as an additional output format in TASverter. A side effect of the implementation is that if your GMM model file can be converted (for instance by TASverter) from its native format to STEP-TAS format, then this viewer can display the geometry. The geometry viewer was an internal development to ease checking of STEP-TAS models and was then considered to be useful for users of TASverter as well. The viewer generates a HTML file with relevant model data and starts a small local web server. The results can then be viewed and 3D zoomed and rotated with any appropriate browser at http://localhost:8080. Optical properties can be overlayed and node ids can be shown.
If at the user's site port 8080 is in use by some other tool then another port can be selected using the `TASVERTER_FLAGS` environment variable, by setting the VIEWER_PORT flag to the port number desired. For instance

```
TASVERTER_FLAGS=VIEWER_PORT=8081
```

will instruct the tool to use port 8081 so the user can view the data at http://localhost:8081. A nice side effect of this feature is that the user can run multiple sessions of the tool in parallel, each with a different port number, and visualise the models in parallel for instance to compare them.

All is handled locally on the user's system without interactions to any external sites or systems.

Note that using the Ctrl-C button to end the viewer process, does not work from the TASverter GUI. In that case the TASverter GUI exit button needs to be clicked which then ends the full session.

TMMverter usage guidelines

The TMMverter extension has been tested extensively which showed that there are too many variations in syntax, often even undocumented, to have them all processed correctly and that the conversion sometimes chokes on syntax that could be interpreted in multiple ways unless additional knowledge of the model data is taken into account. The bulk of the model is converted correctly in our tests so it has been decided to release TMMverter to the public with some guidelines to step-wise improve the conversion.

Here a few guidelines will be given that generally ease the TMM conversion. In principle the user is advised to start with a stripped version of the model, which in general converts without problems, and then gradually add parts that were removed and try to convert again.

- Notes specific to SINDA to ESATAN conversion:
  - From the `HEADER OPTIONS DATA` block remove all lines except the `TITLE` line. Correctly handling these options causes problems and they are not used in ESATAN anyhow.
  - Remove the `HEADER USER DATA` blocks or modify the model such that explicit variables are used via the `HEADER REGISTER DATA` block. The implicitly numbered variables are hard to interpret and detect by the converter.
  - Most unit conversions are done automatically, however, the temperature conversion for now has been switched off. Our testing has shown that it does not fully work correctly as sometimes non temperature quantities are converted which is difficult to notice and correct. This is work in progress on our side.
    In those cases the user is advised to convert the model to SI units using the standard Thermal Desktop features if possible. Another approach could be to replace all initial temperature values by a variable defined in the `HEADER REGISTER DATA` block. This will result in this variable occurring in the ESATAN `$LOCALS` block where it can then be corrected manually. Finally equations and interpolations on temperature will need to be checked and corrected where needed.
- Variables names are kept where possible. Be aware of variable names having a specific meaning in the target tool, for instance KL, QA, QE variables have a special meaning in ESATAN. If those are used in the source SINDA model then changing their name by for instance prepending a Z up front will ease the conversion.
- Copy the model and:
  - Remove radiative conductors and environmental heat loads that were generated with the GMM model, as these would just slow down the conversion due their amount. If this is difficult then perhaps remove the major part of these to speed up the conversion. Once the final conversion is successful the removed items can be added to the input model which is then re-converted in full.
  - Remove user logic in subroutines. These very often contain code and features that are not really needed for the model or that can be represented in some other way in the target tool. Often such code is not written cleanly in a way the converter can process and thus cause difficulty in converting.
- Try to convert the model. If unsuccessful try to remove more from the model data even. Often a failure to convert is caused by the use of a syntax dialect which is not known to the converter. By trying to detect and correct the syntax the problem might be solved. In general an error message is produced which indicates the location and type of error. The user manual of the original tool illustrates the “official” syntax and thus can be used as guideline to determine how to correct things.
- When conversion is successful then add part of the data that was removed previously and try to convert again, see previous step.

- Often a number of these cycles are needed to fully convert the model.
- Note that a small number of subroutines are converted from one tool to the other, sometimes making use of an additional set of subroutines that acts as a translation layer. This set of subroutines is created automatically by TASverter in the current working directory and is meant to be included in the final model file for the target tool.
- Make sure to validate the converted model, for instance by running both the original and the converted model for the same setup and comparing the results. Naturally there will be differences, however, the overall behaviour should be similar. To reduce such differences one could consider to reduce the time step and even enforce a fixed time step in order make the simulations more comparable.

If you have any feedback or questions then please do not hesitate to contact us at tasverter@thermal.esa.int