Model Conversions
Systema vs Step-Tas
Thermisol vs Esatan

Guidelines

September 2024

Page Left Intentionally Blank

Use of the software and of the present software tutorial is submitted to a license agreement to be accepted before the software installation on a computer.

All suggestion or error concerning the software or this software manual can be sent to:

Airbus Defence and Space

For the attention of Systema Team

Z.I. du Palays

31 rue des Cosmonautes

31402 TOULOUSE CEDEX 4

FRANCE

engineering.software@airbus.com

Page Left Intentionally Blank

Table of contents

1   DOCUMENTATION

1.1   Applicable documents

N/A

1.2   Reference documents

N/A

1.3   Normative documents

N/A

Introduction

This document provides guidelines for the model conversions between Systema and other thermal formats and software. Particularly, the conversion of geometrical models between Systema and Step-Tas formats and the conversion of mathematical models between Systema (Thermisol) and Esatan formats will be detailed.

2   SCOPE OF THE STEP-TAS FORMAT

The Step-Tas format handles the definition of a thermal model including the following modules:

-   Network-model and results format (NRF part)
-   Meshed geometric model (MGM part)
-   Space kinematic module (SKM part)
-   Space mission aspects (SMA part)

Currently, the export and import functions of Systema allow the conversion of the meshed geometric model (MGM part) plus the identification of geometrical nodes, the list of materials and material properties as well as elements of the thermal mathematical model (which is part of the NRF).

The mission and kinematics aspects (SMA and SKM parts) are out of the scope of the current Systema converters. Besides, non-geometrical nodes and other network data not related to the geometry are also not included in the conversion.

Results based on a network definition may be exchanged using specific Python scripts developed to convert Systema results – HDF5 files using the Systema Data Storage (SDS) format – to / from Step-Tas (NRF part) format.

3   MANAGEMENT OF GEOMETRICAL AND MATHEMATICAL MODELS IN SYSTEMA

The classical process to generate a complete mathematical model within Systema is as explained in the following diagram.

The complete thermal mathematical model (TMM) may be considered as the output of a process involving Thermica modules (to generate the nodal description, couplings and fluxes). The inputs of this process are the meshed geometrical model, the mission and other additional data usually formatted in the mathematical language, i.e. in Thermisol format.

When dealing with model conversions, it is important to identify the usage that will be made of the exported model and what are the converted data. For example, if the mathematical model shall be re-built (totally or partially) from an exchanged geometrical model, it is important to check that all data required for re-processing the inputs into the new software environment are available.

Inputs / *Outputs*
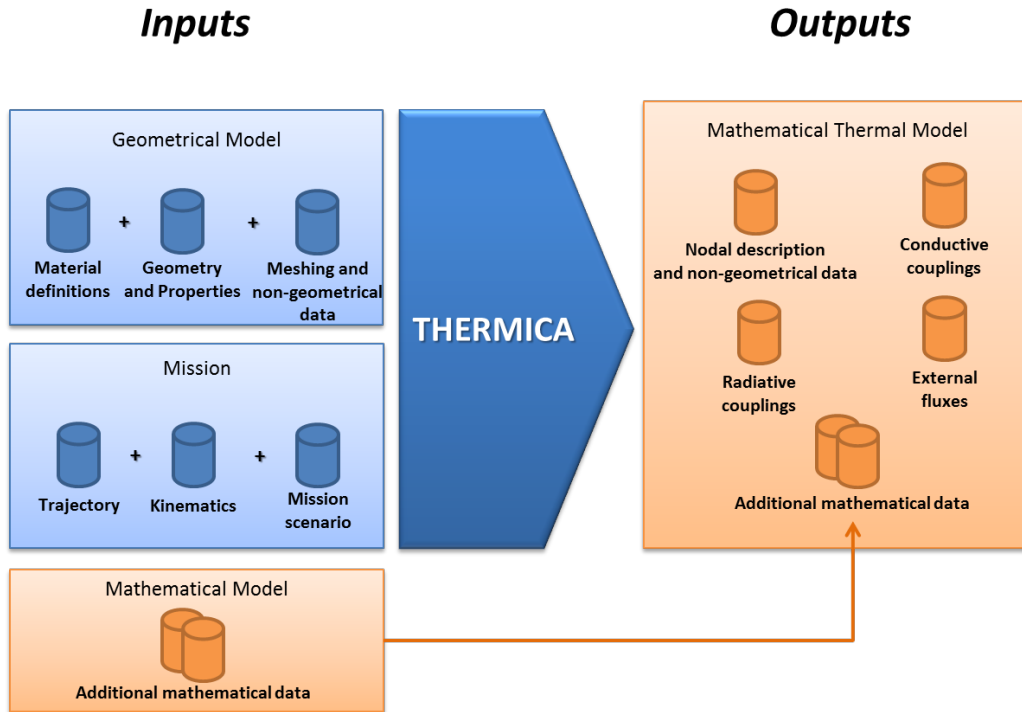
4 CONVERSION USE CASES

The conversion of models through the Step-Tas format concerns mainly the meshed geometric model plus materials and material properties and related network data. This scope may be sufficient to import the geometrical model into another software and run a static radiative couplings computation. However, this is not sufficient to completely be able to retrieve a full mathematical model.

4.1 Meshed geometrical model conversion

The meshed geometric model generally includes a geometrical definition of the model, meshing and numbering parameters, and properties related to materials or not. Different software may have discrepancies in these definitions. The Step-Tas format, which is used for the transfer of these data, also has a limited scope of functionalities. Indeed, the export or import of a Step-Tas file from Systema shall map the data as best as possible between the two formats and report any approximation or impossibility of conversion.

The list of conversion approximations, unsupported features and known bugs is addressed in the next chapter. The limitations may concern:

- Geometrical definitions
- Meshing and numbering options
- Properties
- Specific items

The conversion of models through Step-Tas generally allows performing a nodal description export, radiative couplings and external fluxes computations. However, all additional data (non-geometrical nodes, boundary conditions, internal dissipations, heaters, additional couplings, surface contacts,

edge contact resistances…) are not converted. Conductive couplings may also be difficult to re-process using different software, especially if methodologies are different and if contact resistances have been specified.

All these data, if required, shall then be provided as additional mathematical data, using the proper TMM language.

4.2 Mission conversion

Concerning the trajectory, kinematics and mission aspects, the possibility to map the data between Systema and the Step-Tas formats has not been evaluated. Indeed, it is required to transfer this data in the form of written specifications (plus eventually tabulated files describing time-position-speed or rotation angles) so that it is possible to re-build the correct definitions in another software.

4.3 Mathematical files conversion

The full mathematical model may be converted (see the chapter dedicated on TMM conversion) so to provide a mathematical model reference. More generally, if part of the TMM shall be re-processed (for example by re-computing the radiative couplings or external fluxes), the converted mathematical model may be cleared of these data and re-used to include the newly computed ones. Or eventually, all required additional data written into the mathematical file shall be provided so to be included in a model generated into the new tool environment. In any case, since the geometrical model conversion is generally not sufficient, a partial or full TMM shall be transferred in addition to a Step-Tas file so to preserve the integrity of the thermal modelling.

5 ABOUT THIS GUIDELINES

This document is divided into 3 sections concerning the conversion of:

- Geometrical model: Systema / Step-Tas conversion
- Mathematical model: Thermisol / Esatan or Step-Tas conversion
- Network results: Systema Data Storage / Step-Tas (NRF part) conversion

Each section lists the conformances / approximations / impossibilities of the export and import. Tips may also be given to minimize the risks during the conversions.

These guidelines are written for the last available version of Thermica, i.e. the 4.9.4. If modifications have been made since the 4.5.3 or expected for the near future, it will also be detailed.

Geometrical Model Conversion

The geometrical model conversion concerns the geometrical definition, the meshing and numbering specifications and properties.

6   GEOMETRICAL DEFINITION

The geometry is composed of shapes (shells or volumes) and cutters within a hierarchy of objects.

6.1   Step-TAS validation

The validation of the imported and exported thermal models from STEP-TAS Software Development Kit (SDK) has been integrated since Systema-4.9.0. This validation can be disabled in the settings. It allows get the status of any STEP-TAS file regarding the protocol.

6.2   Objects import / export

The hierarchy of objects is completely imported / exported.

6.3   Shapes export

Only shell shapes are converted (volumes are not). Besides, the Step-Tas format does not handle the following Systema shell shapes: antennas, polygons, extruded polygons, revolved polygons, boxes.

Besides, in versions anterior to the 4.8.0 one, the export of some truncated cones and disc failed.

1. Box 4, 5 and 6 are not supported by Step-Tas. Boxes are divided into rectangles for each face. The properties are applied to each rectangle.
2. CAD shapes are not exported in Step-Tas. You must recreate it with Systema shapes.
3. Only polygons with 3 points bases are exported in Step-Tas. If the extruded vector is not perpendicular to the base, the Step-Tas validation will raise an error but the shape will still be exported.
   If you create polygons with more than 3 points bases, use the functionality to convert the polygon into triangles.
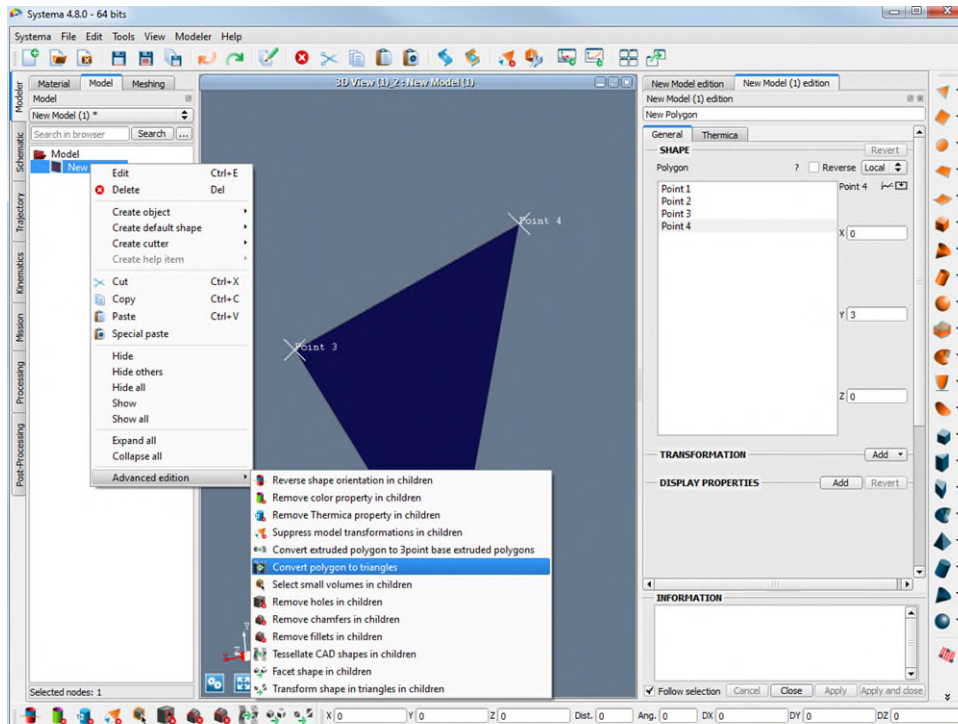
4. The color defined on objects is reported on each Step-tas shapes
5. Specific transformation on objects and shapes are converted into cosines transformation in Step-tas
6. STEP-TAS is very strict when defining quadrangles, an option can be enabled/disabled in the settings to correct the coplanarity automatically before exporting the model in STEP-TAS format. It will apply the same correction as the script "correctQuadrangleDNVersion.py" available in the installation directory of Systema "python\publicScripts\".
7. Antennas are not supported by Step-Tas. Antennas with focal can be exported with Systema 4.9.0 and later. Antennas are divided into a set of triangles.
8. Reverse orientation of shapes in export is taken into account for the following shapes: disc, rectangles, boxes, quadrangles and triangles. Other shapes will be exported as not reversed.

6.4   Shapes import

1. The color of shapes is transferred to their parent object
2. Length and angle are converted in meter and degree. For the other quantities values as SI unit, there is no conversion.
3. Different material for UV and IR thermo-optical properties are not supported by Step-TAS, a warning is raised if different materials are used : "Step-TAS does not support different material for UV and IR thermo-optical properties: a new material has been created."

6.5   Cutters

The list of available cutters in Systema and Step-Tas are different. Systema-4.8.3 manages cylinders and extruded polygons only. Since Systema-4.9.0 infinite solid by plane can be imported (but cannot be created). Both inside cut (Difference) and outside cut (Intersection) can be exported to Step-Tas

---

(the Union operation is not available). "Infinite Difference" cutters are still available for retro-compatibility purposes but it is highly recommended to use finite "Difference" instead to ensure Step-Tas interface robustness.

Management of cutter has been highly improved in Systema-4.9.0. Cutters are no longer replicated when exporting a model.

6.6 Cutter export

1. If you use extruded polygon cutter with more than 3 points base (3 edges polygon cutters), use the functionality to divide them in 3 points base extruded polygon before exporting in Step-Tas. It is necessary otherwise the cutter will not be exported. This kind of cutter does not exist in Step-Tas. In the particular case of 4 points rectangle base with an orthogonal extrusion, the cutter is exported to Step-Tas as a box cutter.
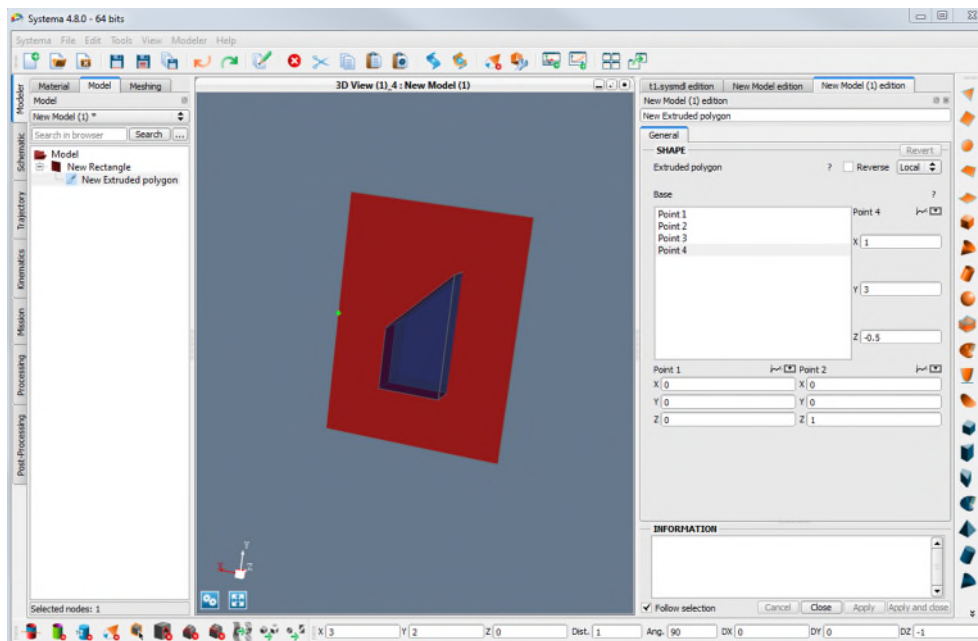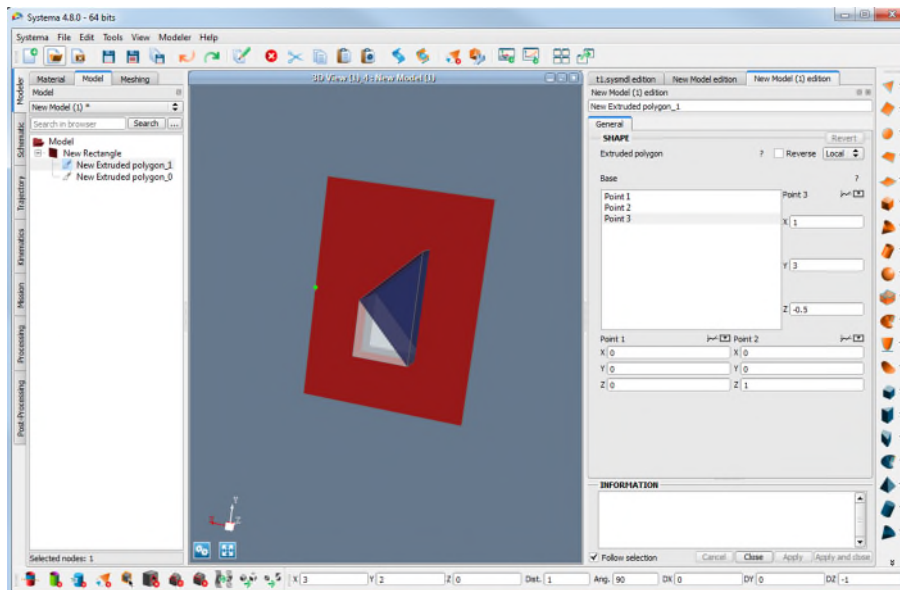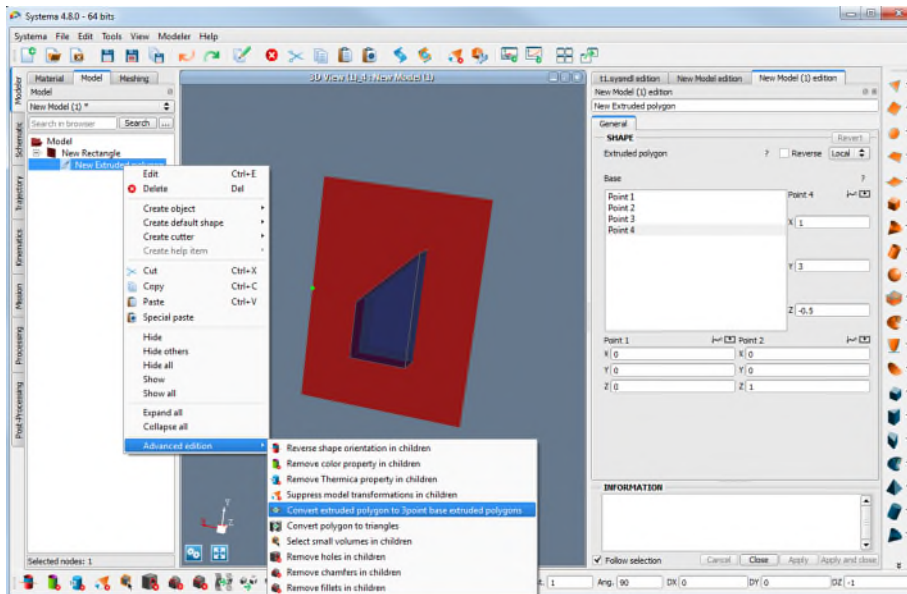
2. Boolean shapes cutting INSIDE (Difference) or OUTSIDE (Intersection) are exported.
In the case of INSIDE cutters, it is highly recommended to use "Difference" cutters instead of "Infinite Difference" cutters because infinite cutters are not managed in Step-Tas.

If you use "Infinite Difference" as it is not handle in Step-Tas it will be exported as "Difference" and in some cases the cutters might not cut your shapes as you expected. To solve this you can select the root of your model and use the option "Convert and extend infinite cutters in children" which will convert cutters from "Infinite difference" boolean operation to "Difference" and extend the extrusion to simulate an infinite cut.
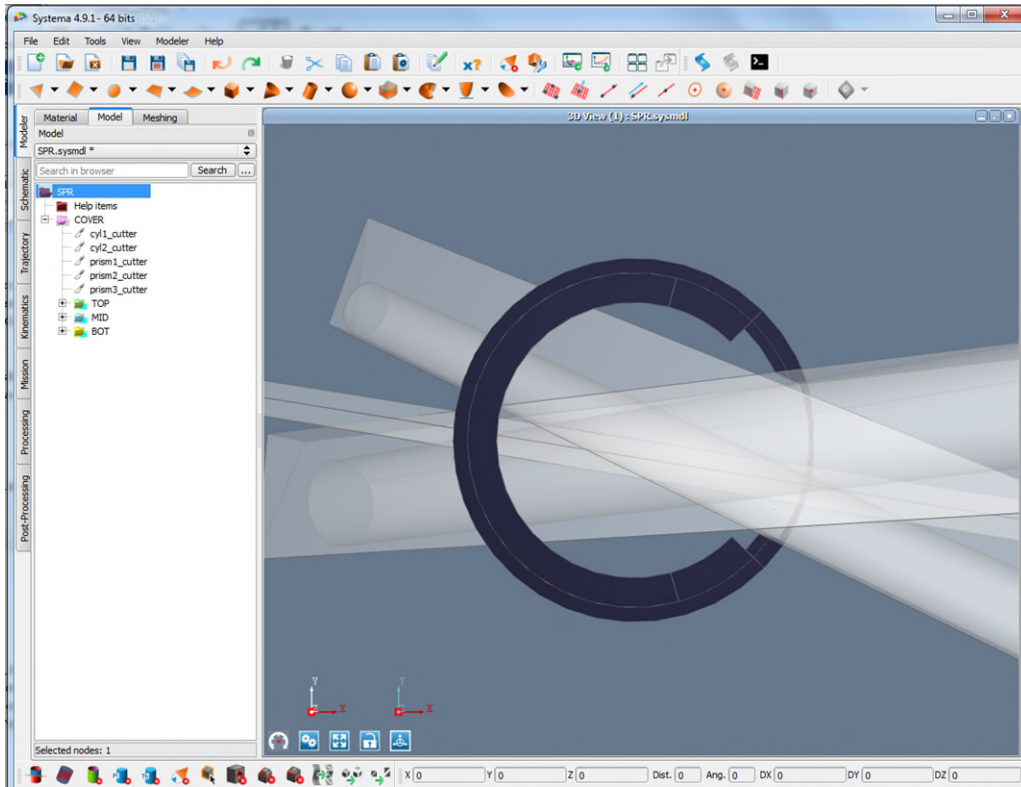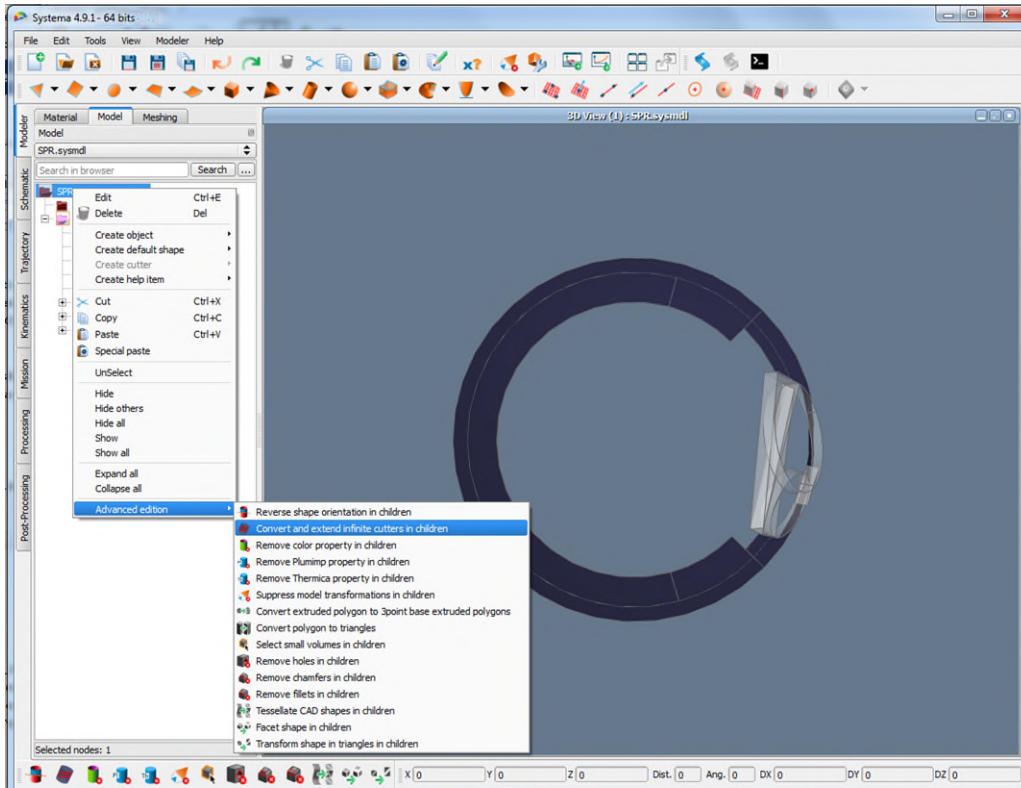
3. Only finite cylinder, box & triangle prism shapes are supported. If an infinite solid by plane cutter was imported it will be exported as an infinite solid by plane cutter as well.

6.7   Cutter import

1. Boolean shapes cutting INSIDE or OUTSIDE are imported as "Difference" or "Intersection" cutters.
2. Infinite solid by plane cutter is managed by extruded polygon cutter that automatically «grows» if shapes are added to the model. This « growing » property is not displayed in the graphical user interface.
3. Transformation on cutters are imported
Beware of the case where shapes and cutters are at the same level in the tree and both hold different transformations. In some cases, the 3D representation of the cut is glitched (see image below).

Two workarounds exist:
1) Resolving one of the transformations (for example the one of the cutter) by selecting "Advanced edition/Suppress model transformations in children
2) Inserting the cutter as a child of the shape to cut and adapt its transformation

* : holds a transformation
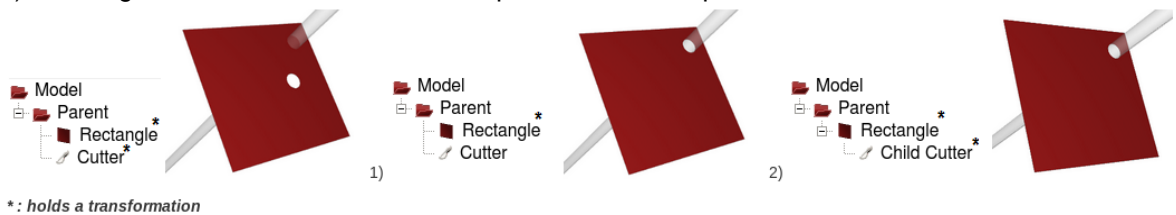
6.8   Material import/export

Materials are imported/exported. Since Systema 4.9.2 phases of materials are supported and sentinel values and temperature dependent conductivity are handled.

7   MESHING AND NUMBERING OPTIONS

Sub-model names are not imported / exported.

Grid spacing and meshing ratios are imported/exported since Systema 4.9.0.

In the specific case of meshed cone import, when the cone diameter 2 is superior to diameter 1, the numbering is not applied in the correct sense. This must be corrected with the script "fixTruncatedConeMeshing.py" available in the installation directory of Systema "python\publicScripts".

8   THERMAL PROPERTIES

Systema-Thermica 4.9.0 includes the following properties (on shells):   *Step-Tas possible mapping*

- Coating
    - o Side management   √
    - o Activity   √
    - o Emission critical level   X
    - o Thermo-optical properties (IR / UV)   √ (but see coating properties)
        - ▪ Emissivity (IR) / Absorptivity (UV)
        - ▪ Specularity (IR / UV)
        - ▪ Direct transmission (IR / UV)
        - ▪ Diffuse transmission (IR / UV)
- Bulk
    - o Side management   X
    - o Activity   √
    - o Thickness   √ (but see bulk properties)
    - o Physical properties   √ (")
        - ▪ Conductivity
        - ▪ Specific heat
        - ▪ Density
- Transverse
    - o Geometrical area correction   X
    - o Transverse conductivity   X
    - o Transverse radiosity   X
- Miscellaneous
    - o Refraction indexes (IR / UV)   X
- Enclosure
    - o Side management   X
    - o Enclosure number   √

8.1 Thermal properties export

Applicative properties inheritance is lost. The properties are reported on each Step-tas shapes.

A new Step-Tas material is created each time the values of applicative properties of a shape are not related to a Systema material.

8.2 Coating properties

The side management in Systema allows specifying properties for both sides or for each positive and negative side. The Step-Tas format manages distinct side properties so they can be imported or exported.

The activity allows turning on / off a shape in the radiative exchanges (including external fluxes). This definition is the same in both formats.

The emission critical level allows selecting different ray specifications at Thermica process level. This option does not exist in Step-Tas. It is then not converted.

Thermo-optical properties have similitudes between Systema and Step-Tas but do not have the exact same definition (the Esatan definition differs also from these two).

In **Systema-Thermica**, the definition of the coefficients is the following:

ε / α : coefficient of emissivity (IR) / absorptivity (UV)

$$diffusivity + specularity + direct\ transmissivity + diffuse\ transmissivity = 1$$

The diffusivity is deduced from the last 3 other coefficients.

This formulae and coefficients definition allows specifying the ratio of reemitted behaviors independently of epsilon or alpha.

In the **Step-Tas** format, the definition of the coefficients is the following:

ε / α : coefficient of emissivity (IR) / absorptivity (UV)

$$\varepsilon\ (or\ \alpha) + reflection + direct\ transmissivity + diffuse\ transmissivity = 1$$

$$reflection = (1 - \varepsilon - direct\ transmissivity - diffuse\ transmissivity) . (diffusivity + specularity)$$

With $diffusivity + specularity = 1$

This means that Step-Tas specularity coefficient is a ratio of specularity vs the reflection. Transmission coefficients are however relative to a global energy conservation law (including absorption and reemission).

If transmission coefficients are equal to zero (which corresponds to more than 95% of the cases), specular coefficients have the same meaning in both formats. Otherwise, a conversion is required.

The usage of thermo-optical coefficients may be unclear if their definitions are not well known, especially because we may use them in different ways (Thermica, Step-Tas and Esatan have three different definitions…). The understanding of the Step-Tas definition has been clarified in the 4.8.0 release of Systema. Previous versions may have an incorrect import / export of thermos-optical properties.

8.3 Bulk properties

Systema-Thermica handles both mono-layer ("Alike side" option) and dual-layers ("Distinct sides" option). This side management is not supported by Step-Tas. The import of Step-Tas models will be correctly imported (using however dual-layer bulks so to set positive and negative thicknesses). However, dual-layers will not be exported to Step-Tas (only the positive bulk material will be kept).

Since Systema 4.9.1 bulk activity is imported and exported following these instructions:

"Whether each side of the primitive is conductively active is determined by both the sizeX_notional_thickness and the sideX_bulk_material attributes, and is also covered implicitly in the STEP-TAS standard. A side is conductively active if its sideX_notional_thickness is greater than zero and its sideX_bulk_material is set. It is conductively inactive if its thickness is unset or zero AND its bulk material is also unset."

The thickness is converted to Step-Tas using positive and negative values. For Systema mono-layer bulk shapes, the thickness is divided by two at export. Zero thickness values are exported to Step-Tas. In this case, an error is raised.

8.4 Enclosures

Enclosures are imported / exported since Systema 4.9.1.

The enclosure container may be used to define different enclosures onto a model. The default enclosure, 0, shall always correspond to the external or total part of the spacecraft. Enclosures 1 to 99 may be used to define internal cavities.

It is possible to define a discrete list of enclosures for a shape. Exemple of a list of enclosures (for a shape belonging to 3 different ones for example): "0 96 97".

At processing level, it will then be possible to filter the radiation computation for a specific enclosure.

Defining enclosures will be useful in the case of a full model composed of internal and external parts where moving bodies are defined in the external part (typically moving solar arrays). Then, using different radiative couplings computations for each module will optimize the computation time since only the external enclosure would lead to dynamic computation (i.e. for each orbital position) while the internal enclosures would be considered as static (i.e. without time dependencies).

8.5   Other properties

Miscellaneous, transverse properties are not supported by the import / export with Step-Tas.

9   SUMMARY OF STEP-TAS (MGM) IMPORT / EXPORT LIMITATIONS

Actions	4.5.3	4.6.1	4.7.1	4.8.0	4.8.3	V4.9.0	V4.9.1	V4.9.x	Comment
General									
Update of the precision as "1.0E-5" instead of "1.0E-07"	✘	✘	✔	✔	✔	✔	✔	✔	*Correction of accuracy consistency in Step-Tas file*
Import / export of parametric data (using variables)	✘	✘	✘	✘	✘	✘	✘	✘	*Import of parametric models not implemented* *Export of valued parameters*
Units management	✔	✔	✔	✔	✔	✔	✔	✔	*Systema only handles SI units. The conversion of the geometry is however correct.*
Geometry - shapes									
Export of truncated cone/disc	✘	✘	✘	✔	✔	✔	✔	✔	*Correction of some export bugs*
Export of truncated paraboloid shapes	✘	✘	✘	✘	✘	✘	✘	✘	*Not supported by Step-Tas*
Export of antenna shapes	✘	✘	✘	✘	✘	✔	✔	✔	*Split into triangles*
Export of CAD shapes	✘	✘	✘	✘	✘	✘	✘	✘	*Not supported by Step-Tas*
Automatic export of Box 4, 5 and 6	✘	✘	✘	✔	✔	✔	✔	✔	*Split into rectangles*
Export of polygons	✘	✘	✘	✔	✔	✔	✔	✔	*See procedure described above for restrictions*
Export of extruded polygon cutters	✘	✘	✘	✔	✔	✔	✔	✔	*See procedure described above for restrictions*

Actions	4.5.3	4.6.1	4.7.1	4.8.0	4.8.3	V4.9.0	V4.9.1	V4.9.x	Comment
Import / Export of extruded polygons	✘	✘	✘	✘	✘	✘	✘	✘	*Could be managed by splitting but not implemented*
Import / Export of revolved polygons	✘	✘	✘	✘	✘	✘	✘	✘	*Could be managed by splitting but not implemented*
Import / Export of solid shapes	✘	✘	✘	✘	✘	✘	✘	✘	*Step-Tas definitions of solids not sufficient*
Automatic correction of quadrangles coplanarity on export	✘	✘	✘	✘	✘	✔	✔	✔	
Geometry - transformations									
Import of rotations with fixed axis	✘	✔	✔	✔	✔	✔	✔	✔	*Correction of bug at import of these transformations*
Export of transformations	✔	✔	✔	✔	✔	✔	✔	✔	*Converted to cosines transformations*
Export of "reversed" shapes	✘	✘	✘	✘	✘	✔	✔	✔	*Handled for triangles, rectangles, discs, boxes and quadrangles. Other shapes are exported as not reversed.*
Geometry - cutters									
Import / export of outside cutters	✘	✘	✘	✘	✔	✔	✔	✔	*Intersection cutters*
Import / export finite boolean shapes	✘	✘	✘	✘	✔	✔	✔	✔	*Difference cutters*
Import /export of transformation on cutters	✘	✘	✘	✘	✔	✔	✔	✔	*Beware of the problematic case described above*
Import / export of half-space cutters	✘	✘	✘	✘	✘	✔	✔	✔	
Import of truncated cylinder cutters	✘	✘	✘	✘	✘	✔	✔	✔	

Actions	4.5.3	4.6.1	4.7.1	4.8.0	4.8.3	V4.9.0	V4.9.1	V4.9.x	Comment
Import of infinite solid by plane cutters	✘	✘	✘	✘	✘	✔	✔	✔	*Managed by extruded polygon cutter that automatically « grows »*
Import of cone, sphere, parabola cutters	✘	✘	✘	✘	✘	✘	✘	✘	*Not available in Systema*
Meshing									
Import / export of meshing ratios	✘	✘	✘	✘	✘	✔	✔	✔	
Import / export of sub-model	✘	✘	✘	✘	✘	✘	✘	✘	*May be implemented if need be*
Import / export of numbering	✔	✔	✔	✔	✔	✔	✔	✔	*Exported as a list (rules to generate numbers lost)*
Properties									
Export of properties (general)	✔	✔	✔	✔	✔	✔	✔	✔	*Inheritance lost at export (but local values on shapes)*
Import of colors	✔	✔	✔	✔	✔	✔	✔	✔	*Restricted to object level*
Import / export of Thermo-optical coefficients	✘	✘	✘	✔	✔	✔	✔	✔	*Correction made on exported Step-Tas coefficients*
Optimization of material export	✘	✘	✘	✔	✔	✔	✔	✔	*Avoid duplication of materials definition when local values are used in Systema*
Import / export of dependent properties	✘	✘	✘	✘	✘	✘	✘	✘	*Temperature, time, incident angle, wavelength dependencies are not managed by Step-Tas*
Export of multi-phases material properties	✘	✘	✘	✔	✔	✔	✔	✔	*Default material phase exported until 4.9.1. Full support since Systema 4.9.2*
Import of multi-phases material properties	✘	✘	✘	✘	✘	✘	✘	✔	

Actions	4.5.3	4.6.1	4.7.1	4.8.0	4.8.3	V4.9.0	V4.9.1	V4.9.x	Comment
Export of specific item (non-geometrical nodes, additional couplings, dissipations, heaters, surface contacts…)	✗	✗	✗	✗	✗	✗	✗	✗	*Not managed by Step-Tas. However specific items may be transferred as additional data written in the solver language*

✗ : Not supported

✓ : Supported with restrictions

✓ : Fully supported

10 SYSTEMA TO STEP-TAS TMM EXPORT

As of version 4.9.4 of Systema, a part of the TMM can be exported with the geometrical model into a STEP-TAS file.

The STEP-TAS converter processing is used to export data from the thermal mathematical model (TMM) into a STEP-TAS file. The exported data are:

- node properties:
    - node name
    - capacitance
    - internal dissipation
    - initial temperature
- radiative couplings
- conductive couplings

In order to proceed, this converter needs 3 parameters.

- two input parameters being the GMM (geometrical model) and the Thermisol mathematical model.
  The geometrical model is in the STEP-TAS format while the mathematical model is in thermisol dck format.
- one output parameter being the newly created TMM.
  The single TMM file is in the STEP-TAS format and is the result of the completion of the input GMM with TMM data from Thermisol.

10.1 Command line use

The use of the DCK to STEP-TAS converter in command line is is done through the following command:

Dck2Steptas[WIN.exe] -gmm <your_gmm_STEP_file> -dck <your_dck_tmm_file> -out <output_tmm_STEP_file>

Please note that both DCK and GMM files have to be related. Especially, the DCK file must be generated from processings based upon the GMM meshing.

10.2 HMI process tab use

Systema also provides the DCK to STEP-TAS conversion from the processing tab.

This processing is available under the Thermisol section and is displayed as:
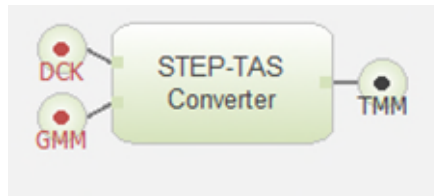
The parameters are exactly the same as the command line use. The dck input can however be the result of other processing boxes like bellow.
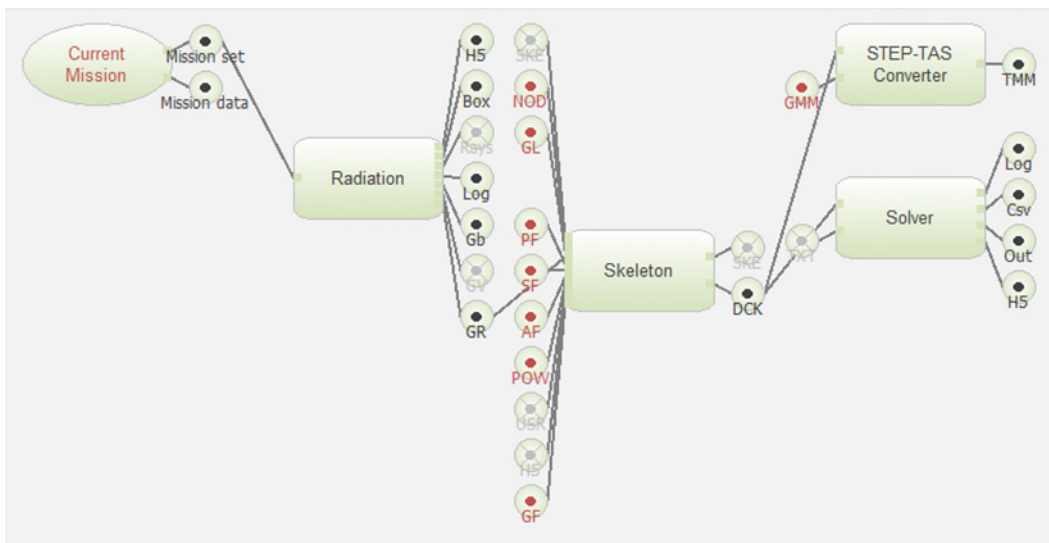
⚠ The STEP-TAS geometrical model is not available from the mission. Hence, the user has to manually export the meshing and complete the GMM input of the processing before the run.

📋 It is possible to use the resulting STEP-TAS file with the TASVerter tool from ESA. In order to have a compatible file, please modify the following line:

        NRF_TOOL_OR_FACILITY('TMMverter','Systema vs STEP-TAS converter',$);

into

        NRF_TOOL_OR_FACILITY('TMMverter','ESATAN vs STEP-TAS converter',$);

11 SYSTEMA TO STEP-TAS EXPORT REQUIREMENTS

A previous list of requirements has been written by Airbus DS thermal engineers concerning thermal building with Systema-Thermica to ensure the compatibility with Step-Tas. This requirement list has been done for the 4.5.3 version of Systema.

This requirement list is updated to take into account recent evolutions until the 4.9.0 release. In case the applicable version of Systema is still the 4.5.3 one, it is still possible to use the 4.9.0 in order to export the model in the Step-Tas format. This is even recommended and added as a new export requirement.

- ➢ **Requirement STEP-EXP-0:** The version 4.9.0 shall be used for exporting the geometrical model.
- ➢ **Requirement STEP-EXP-1:** No truncated paraboloid shape shall be used.
- ➢ **Requirement STEP-EXP-2:** No antenna with focus shape shall be used.
- ➢ **Requirement STEP-EXP-3:** Polygon shapes shall be split using the dedicated Systema option.
- ➢ **Requirement STEP-EXP-4:** No extruded and revolved polygons shall be used.
- ➢ **Requirement STEP-EXP-5:** No CAD model shape shall be used.
- ➢ **Requirement STEP-EXP-6:** No volumetric mesh shall be used.
- ➢ **Requirement STEP-EXP-7:** Avoid the use of "Infinite Difference" cutters. If used, their length shall be sufficient so to avoid confusions between finite / infinite definitions. Polygon cutters should be split using the dedicated Systema option (as explained above).
- ➢ **Requirement STEP-EXP-8:** Two-layers bulk shall not be used (implement two distinct shapes instead).
- ➢ **Requirement STEP-EXP-9:** If transverse properties are defined, the corresponding couplings shall be implemented in the TMM.
- ➢ **Requirement STEP-EXP-10:** It is preferred to declare all thermo-optical properties in a material file.
- ➢ **Requirement STEP-EXP-11:** No temperature or other dependency shall be used.
- ➢ **Requirement STEP-EXP-12:** No non-geometric node or any other specific item shall be used, they shall be defined in the TMM.

12 STEP-TAS TO SYSTEMA IMPORT REQUIREMENTS

The preliminary 4.5.3 list written by Airbus DS thermal engineers has also been upgraded to fit the 4.9.0 evolutions. This requirement list concerns Step-Tas features which are not fully compatible with Systema. The integrity of the Step-Tas model versus the native one is of the responsibility of the model provider. A general requirement may however be made concerning the exclusive usage of Step-Tas compatible features when implementing the native model.

- ➢ **Requirement STEP-IMP-0:** The version 4.9.0 shall be used for importing the geometrical model. If a previous version of Systema has to be used, the imported model from the 4.9.0 can be exported in the 4.5.3 format (LTS export)

- ➢ **Requirement STEP-IMP-1:** The native model shall exclusively use Step-Tas compatible features excluding the limitations expressed in further requirements. The model may however be completed by a TMM (for additional data and user logic) and/or written specifications to be agreed (for kinematics bodies…)

- ➢ **Requirement STEP-IMP-2:** Boolean cut operation can only be of cylinder, box or triangle prism types.

- ➢ **Requirement STEP-IMP-3:** No solid shall be used.

- ➢ **Requirement STEP-IMP-4:** No parametric data or properties shall be used.

- ➢ **Requirement STEP-IMP-5:** No non-geometric node or other additional non-geometrical data shall be used, they shall be defined in the TMM.

- ➢ **Requirement STEP-IMP-6:** No moving body shall be used. If needed, a written specification shall be provided in complement of the Step-Tas model.

Mathematical Model conversion

These guidelines provide information and tips about the Esatan / Thermisol compatibility and conversions. Other formats that may be required are MSC/Sinda and Sinda/Fluint. A dedicated section to these formats should be written in the future.

The Thermisol language is based on Esatan but both tools have developed through the years different features. This section describes the differences between the two formats and details the Thermisol-Esatan converter functionalities.

13 EXCLUSIONS

The FHTS extension of Esatan is not supported by Thermisol and cannot be converted.

The Power and Outgassing add-ons of Thermisol are not supported by Esatan and cannot be converted.

Wavelength dependencies are managed differently in Thermisol and in Esatan. There is no available converter between the two languages to support this feature. However, wavelength dependent data can be generated from the geometrical model using Thermica or Esatan-TMS.

14 ESATAN TO THERMISOL

Thermisol has been developed to support the Esatan language. Except for the FHTS part, any Esatan input should be able to run with Thermisol. Some very specific Esatan functions or variables without any impact on the thermal model may be automatically ignored (this will be mentioned in the log output of Thermisol). Otherwise, all control variables and Esatan functions should be available in Thermisol (*if not, please report this to us*).

However, when running with Thermisol a model written for Esatan, it is possible that the pre-processor complains about some warnings and errors. In fact, both software have different syntax checker and may not raise the same mistakes. In that case, the Thermisol log output should explicit the line number and the type of error found. For example, Thermisol may raise an error if a variable in read access has not been declared or valued before, or if an array has more or less elements than the declared size. Normally, errors can be sent whenever there is a possible mistake in the input file. Those need to be corrected so to be able to run the Thermisol solver.

15 THERMISOL TO ESATAN

The Thermisol language, even if compatible to the Esatan one, has been upgraded through the years in order to answer specific modelling needs and to ease the writing of a thermal model. Generally, a model written for Thermisol will fail if used as it is with Esatan. A converter is then necessary.

The Thermisol to Esatan converter can be accessed using the following Thermisol batch commands:

```
ThermisolWIN.exe –c model.dck
```
(on Windows)

```
ThermisolLNX –c model.dck
```
(on Linux)

This will create a *model.tan* file compatible with Esatan.

There is no need for a valid license to call the converter so the functionality is free.

The conversion will concern the following:

- File and data format
- File and blocks structure

The corrections or incompatibilities are listed in the next paragraphs.

15.1   File and blocks structure

The structure of the file and blocks differs between Esatan and Thermisol. Here is a summary of all differences.

Thermisol structure	Esatan structure

$GLOBAL[*-n*]
  $LOCAL[S]
  $VARIABLE**[S]   or   **$CONSTANT[S]
  $ARRAY[S]
  $TABLE[S]
$ENDGLOBAL	

$MODEL *name*
  [$MODEL ... ENDMODEL]
  $ENTITITIES
  $NODE[S]
  $EDGE[S]
  $CONDUCTOR[S]
  $EVENT[S]
  $LOCAL[S]
  $VARIABLE**[S]   or   **$CONSTANT[S]
  $CONTROL[S]
  $ARRAY[S]
  $TABLE[S]
  $SUBROUTINE[S]
  $OUTPUT**[S]   } *or* { **$VARIABLES1
  $VTEMPERATURE**[S]   **$VARIABLES2
  $VTIME[S]
  $VRESULT[S]
  $PARAMETER[S]
$ENDMODEL
	$MODEL *name*
	  [$MODEL ... ENDMODEL]
	  $USER_NODE_ENTITIES
	  $NODES
	
	  $CONDUCTORS
	  $EVENTS
	  $LOCALS
	  $CONSTANTS
	    $CONTROL
	  $ARRAYS
	      $TABLE
	  $SUBROUTINES
	  $OUTPUTS
	
	  $VARIABLES1
	  $VARIABLES2
	
	  $PARAMETERS
	$ENDMODEL

➢ **Block names with [S]**

Thermisol understands block names with or without ending S characters.

The converter automatically adds or removes S characters to match the Esatan definition.

➢ **$GLOBAL…$ENDGLOBAL**

The $GLOBAL…$ENDGLOBAL structure allows specifying declarations of constants, variables, arrays and tables which are global to the model and all sub-models. This is mainly used to define material properties which may be needed by any model without having to duplicate their declarations. This is particularly required when parametric analysis shall be done on material properties if sub-models are used, avoiding mistakes on multiple declarations of global parameters.

The converter automatically adds all declared data from the Thermisol $GLOBAL block to all models.

➢ **$GLOBAL**[-*n*]**…$ENDGLOBAL**

An optional integer tag may be specified on the $GLOBAL structure. This allows defining different sets of values of the declared data. This feature is mainly used for outgassing problems for which many predefined cases have to be computed without the need of duplicating the full mathematical model (the case selection is done at computation level by specifying the case number on the command line).

The converter is not compatible with this feature. Only one $GLOBAL case shall be kept before converting the model.

➢ **$VARIABLES** or **$CONSTANTS**

The default Thermisol name for the declaration of variables is $VARIABLES. This modification has been made so to have a better consistency between the naming of this block and its purpose (otherwise it is necessary to explain the $LOCALS is used to define constants and $CONSTANTS is used to define variables). However Thermisol understands both syntaxes.

The converter renames $VARIABLES blocks with $CONSTANTS.

➢ **$CONTROL**[**S**]

In Thermisol the $CONTROL block has the same hierarchy level than $VARIABLES (or $CONSTANTS). In Esatan, this block is a sub-element of the $CONSTANTS one.

The converter moves the $CONTROL block so it is part of the $CONSTANTS one (or add a $CONSTANTS block before getting into the $CONTROL one).

➢ **$TABLE**[**S**]

This block is also an independent one in Thermisol. For Esatan, it is a sub-element of the $ARRAYS block.

The converter moves the $TABLE block so it is part of the $ARRAYS one (or add a $ARRAYS block before getting into the $TABLE one).

➢ **$EVENTS sub-elements**

This block has similar definition between Thermisol and Esatan. The main difference concerns periodic events. Thermisol can define periodic events using a modulo style definition (for example REAL* MyEvent = value[period]). Periods may also be defined into a $PERIOD sub-element of the $EVENTS block.

The converter moves all $PERIOD definitions to a $LOCALS block (or create one). Periodic definitions shall however be manually corrected by the user.

➢ **$LOCALS, $VARIABLES (or $CONSTANTS), $ARRAYS, $TABLES sub-elements**

In these blocks, the type of declared data may be specified using $INTEGER, $REAL or $CHARACTER sub-blocks or by explicitly preceding the data definition by INTEGER*, REAL*, or CHARACTER*. In this last case, the * character is not mandatory in the Thermisol format.

The converter automatically adds * characters if missing.

➢ **$ENTITIES**

The $ENTITIES block allows specifying new nodal entities for the current model or sub-model. These entities may be used as any other default ones (i.e. accessible in all blocks, in particular into $NODES, and in all functions receiving entities as an input).

The converter will automatically rename the block $USER_NODE_ENTITIES. This will be correct if no sub-model is used (otherwise a manual correction to set $USER_NODE_ENTITIES on the main model only is required). Besides, all declarations of user defined entities within the $NODES block are moved to the $INITIAL block since Esatan does not support this feature.

➢ **$EDGES**

This block was made to declare edges used by the RCN conductive couplings computation. The advantages of declaring these conductive interfaces into a separate block are that it allows the definition of edge/node connectivity (so conductive fluxes computation routines such as FLUXL, FLUXGL… may be called using classical node groups, edges involved in the conductive flux being automatically taken into account), and it also limit the complexity of the mathematical model to be solved (edges being temperature variables like classical nodes but without any mass, capacitance, radiative or external fluxes – their temperatures can then be computed with optimized numerical methods).

The converter will automatically replace the edge definitions by nodes of an EDGE sub-model. Edge identifications are also modified to their node id correspondences. However, if fluxes computation routines are used (such as FLUXL, FLUXT, FLUXGL, FLUXGT, FLUXML, FLUXMT…), conductive fluxes will not be correctly computed. In that case, a manual correction is required.

➢ **$VTEMPERATURE / $VTIME / $VRESULT   vs $VARIABLES1 / $VARIABLES2**

The new Thermisol executive blocks allow a better numerical integration of user's logic. It makes a clear split between temperature and time dependencies that shall be taken into account for steady-state and transient temperature integration routines.

The converter replaces $VRESULT by $VARIABLES2 which is completely equivalent. $VTEMPERATURE is replaced by $VARIABLES1 and $VTIME will also be included into this block but within a logical condition checking if the temperature integration routine calling this block is a transient one.

*We remind that in transient cases, time dependencies **have to** be called before the resolution of a time-step, so has to be written in $VARIABLES1 (if placed into $VARIABLES2, there will be a numerical shift of one time-step concerning all time dependencies). Moreover, temperature dependencies are not solved correctly in transient cases if $VARIABLES1 is used: the estimation of fluxes at time n+1 will be based on temperatures computed for the time n (so with a numerical shift of one time-step). As a consequence, the two executive blocks definitions are not equivalent and the time-step for the temperature integration may*

need to be decreased so to minimize the effect of numerical shifts induced by the $VARIABLES1 / $VARIABLES2 definition.

➢ **$PARAMETERS**

The Thermisol definition of this block corresponds to a quite old version of Esatan (in which evolutions have been made). Since this block is generally never used (parametric analysis being handled by upper-level scripts or processes), there hasn't been any modification in Thermisol.

The converter will keep this block "as is" so corresponding to an old Esatan format and might not be compatible to recent versions. This block should then be avoided.

15.2 File and Data Format

➢ **Free format:**

Thermisol uses free format without any column restrictions (which is also able to understand strict Fortran77 formatting).

The converter reformat all executive lines so to set tags on the first 5 columns, set continuation line characters on the 6th column, start instructions at the 7th column, and end lines before the 128th column.

➢ **Semicolon characters:**

Thermisol does not require semicolon at the end of declarative lines. Semicolons at the end of executive lines are also ignored.

The converter adds missing semicolons of declarative lines and removes them in executive lines.

➢ **Variable and subroutine names:**

Thermisol has no limitation on name lengths.

If variable or subroutine names exceed 16 characters, they are automatically renamed VARn (truncated names could lead to mistakes). The log file of the conversion lists all renamed variables and subroutines.

➢ **Variables and loops in declaration blocks**

Thermisol can use Fortran style loops in the declaration blocks. Also variables used in these loops or more generally in the $NODES or $CONDUCTORS blocks may have any name.

The converter renames all variables within the $NODES and $CONDUCTORS block with KLn names. The definitions of the loops within these blocks are also reformatted to the Esatan style.

➢ **Double precision management**

Thermisol automatically interprets any floating value as a double precision one (even if written as a single precision one). Also, it replaces any REAL declaration in subroutines by DOUBLE PRECISION ones.

The converter corrects the format of all floating values using a D exponential. Native Fortran functions (such as LOG, EXP, SQRT, COS, ACOS,…) are automatically converted to their double precision equivalents (DLOG, DEXP,…).

All REAL declarations in subroutines are changed to DOUBLE PRECISION.

> **Logical operands:**

Thermisol understands the C style logical operands <,<=,>,>=,==,!=,&&,||.

The converter converts the C style logical operands into Fortran style.

> **Events related logical operands:**

Thermisol uses direct syntax such as BEFORE event, AFTER event, AT event, BETWEEN event1 AND event2.

The converter replaces these instructions with the corresponding Esatan syntaxes.

> **Control Constants:**

Some control constants are specific to Thermisol: ENBALT, INBALT, ERRMIN, ERRMAX, H5_NODES, H5_FREQ, H5_RES0, H5_RES1, CSV_FREQ.

The converter replaces CSV_FREQ by the Esatan PRGFBK control. Others are removed.

> **Thermisol dump functions:**

The converter removes calls to the H5_INIT and H5_DUMP functions. For using Esatan dump function, a manual correction is required.

> **Crank-Nicholson algorithm:**

The Crank-Nicholson routine name is SCRANK. Esatan developed latter a Crank-Nicholson routine but named SLCRNC. Thermisol understands both.

The converter will replace CALL SCRANK by CALL SLCRNC.

> **Mortran subroutine SPLIT option:**

The SPLIT option allows splitting automatically a Mortran subroutine so to avoid compilation issues.

The SPLIT option is removed by the converter.

> **Mortran syntax: Nxxxx**

Thermisol can directly address the internal identification of a node through this syntax, using the nodal entity 'N'.

The converter translates this expression using the INTNOD function.

> **Mortran syntax: NSxxxx**

Thermisol can read / modify a nodal status directly by using the nodal entity 'NS'.

If the node is explicit (written as a number), the converter translates this expression using the STATST function.

If the node is implicit (written as an expression or a variable), a warning is raised and a manual correction is required.

> **Mortran syntax: GLS(xxxx, yyyy)   GRS(xxxx, yyyy)   GFS(xxxx, yyyy)**

As for the nodal status, coupling status may also be accessed directly using this syntax.

A warning is raised and the converter asks for a manual correction (using STATRP functions).

> **Mortran implicit syntax with integer variables or expressions:**

Thermisol allows the access of nodal entities or couplings with the use of variables or expressions. For example, the temperature of a node may be written T:expression, or a coupling between two nodes may be expressed GL(expression1,expression2).

In executive blocks and for nodal entities, the converter will translate these syntaxes using INTNOD functions so to access to nodal entities vectors.

For implicit nodal entities used in declaration blocks, the converter will raise a warning and the need of a manual correction.

For implicit couplings, the converter will also raise a warning and the need of a manual correction.

➢ **Mortran implicit syntax with string syntax:**

In addition to integer variables, the identification of a node may be given through a string variable. This eases the use of sub-models because nodal identifications may be stored into string variables suitable for the identification of any node.

In that case, the converter will behave as above, raising a warning inviting the user to check the conversion. In fact a manual correction is required.

➢ **Mortran macro syntax using znodes strings:**

*Thermisol allows also the assignment for a nodal entity or a coupling of a group of nodes (for couplings, only on the second node id). For example, it is possible to write T:'#100-200' = 0.0 or GR(300,'ALL') *= 0.9. The operands may be = or *=.*

For nodal entities and the = operand, the converter will translate the syntax using the SETNDR function.

In other cases, the converter will expand the macro using as many lines as required (i.e. the number of nodes in the group specified by the znodes string).

15.3   List of Thermisol routines

Here is a list of all Thermisol routines and there compatibilities with Esatan.

Possible values are **OK** if fully compatible, **CONV** if managed by the converter, **POK** if partially ok, **KO** if not compatible

- **Solution routines**
  - SOLVIT   **OK**
  - SOLVFM   (*uses a Newton-Krylov algorithm*)   **OK**
  - *SLFWBK*   (*calls SCRANK, still available via SLFWBK_OLD*)   **OK**
  - *SLFWBK_OLD*   **KO**
  - *SLCRNC*   (*calls SCRANK*)   **OK**
  - SCRANK   **CONV**
  - SCRANKAUTO   **KO**
  - SOLCYC   **OK**
  - SCYCLE   **KO**
  - SCYCLE_ADDSPEC   **KO**

- **Heat transfer routines**

- HEATBAL **OK**
- FLUXL **OK**
- FLUXR **OK**
- FLUXF **OK**
- FLUXT **OK**
- FLUXGL **OK**
- FLUXGR **OK**
- FLUXGF **OK**
- FLUXGT **OK**
- FLUXML **OK**
- FLUXMR **OK**
- FLUXMF **OK**
- TSINK (*original version*) **OK**
- TSINK (*calls TSINKN – Astrium version*) **KO**
- TSINKN **KO**

- **Interpolation routines**
  - INTRP1 **OK**
  - INTRP2 **OK**
  - INTRP3 **OK**
  - INTERP **OK**
  - INTRPA **OK**
  - INTCY1 (*returns automatically average value in Steady-State*) **POK**
  - INTCY2 **OK**
  - INTCY3 **OK**
  - INTCYC (*returns automatically average value in Steady-State*) **POK**
  - NODFNC **OK**
  - CNDFNC **OK**

- **Nodal network management routines**
  - STATST **OK**
  - STATRP **OK**
  - MODELON **OK**
  - MODELOFF **OK**
  - SAVET **OK**
  - FETCHT **OK**
  - INTNOD **OK**
  - NODNUM **OK**
  - SUBMDN **OK**
  - SUBMOD **OK**
  - INTGL **OK**
  - INTGR **OK**
  - INTGF **OK**
  - SETNDI **OK**
  - SETNDR **OK**
  - SETNDZ **OK**
  - GRPMIN **OK**
  - GRPMAX **OK**
  - GRPSUM **OK**
  - GRPAVE **OK**
  - GRPAVE2 (*quadratic average value*) **NOK**

- GRPAVE_W	(*weighted average*)	**NOK**
- GRPAVE_W2	(*quadratic weighted average value*)	**NOK**

- **Data output routines**
  - PRNDTB — **OK**
  - PRHEAD — **OK**
  - PRNDBL — **OK**
  - PTSINK — **OK**
  - PTSINKN — **NOK**
  - QRATES — **OK**
  - PRQBAL — **OK**
  - PRQBOU — **OK**
  - PRQNOD — **OK**
  - PRTTMD — **OK**
  - DMPTHM — **OK**
  - PRTNSM — **OK**
  - PRTNAV — **OK**

- **H5 output routines**
  - H5_INIT — **CONV**
  - H5_DUMP — **CONV**
  - H5_CLOSE — **CONV**

- **Mathematical routines**
  - INTGL1 — **OK**
  - INTGL — **OK**
  - THRMST — **OK**
  - THSTAT (*other thermostat routine*) — **NOK**
  - THSTAT2 (*other thermostat routine*) — **NOK**
  - ADIM — **OK**
  - ADIMVL — **OK**
  - ASIZE — **OK**
  - ACONST — **OK**
  - AELCPY — **OK**
  - AELADD — **OK**
  - AELSUB — **OK**
  - AELMLT — **OK**
  - AELDIV — **OK**
  - AELINV — **OK**
  - AELGT — **OK**
  - AELGE — **OK**
  - AELLT — **OK**
  - AELLE — **OK**
  - AELEQ — **OK**
  - AELNE — **OK**

- **Others**
  - ZDAYDT — **OK**

- ZDAYTM   **OK**

16 ESATAN / THERMISOL CONVERSION REQUIREMENTS

Since Esatan and Thermisol are being very closed in their structures and syntaxes (Thermisol being able to read Esatan files and to re-format mathematical models into Esatan style), there isn't any strong restriction or limitation on the implementation of mathematical models regarding the syntax or data formatting. However, recommendations and processes for checking the quality of models may be respected.

16.1 General requirements

These concern mathematical models implemented in Esatan or Thermisol.

➢ **Requirements TMM-GEN-1:** Limit the user's logic (additional code) to the strict minimum on exchanged models. It is so required to remove all dead code, parametrisation, output routines or whatever is dedicated to internal usage and not related to the thermal modelling.

➢ **Requirements TMM- GEN-2:** All additional code shall be written with respect to standard coding rules. For example, the following rules shall be checked:

o Declare all variables (so no implicit variables, except for loop indexes)

o Comment the code to make it cleaver with an appropriate ratio of codes vs comments (not too few, not too much)

o Use a comprehensible indentation of the code (and comments shall be indented as well) and do not mix spaces and tabulations (that might lead to incorrect indentations in different text editor)

o Write understandable statements. Especially, do not cut a line in the middle of a symbol

o Do not use any COMMON block but use Mortran subroutines and arguments

➢ **Requirements TMM- GEN-3:** Avoid the use of internal solver structures; especially Fortran routines starting with the definition of a COMMON block corresponding to the solver's internal data-structure **shall never be used.**

➢ **Requirements TMM-GEN-4:** When reusing code from a previous model, it shall be checked that it is valid in the new model context.

➢ **Requirements TMM-GEN-5:** Any pre-compiled subroutine shall be given as source code in the mathematical model so to avoid compiler compatibility problems.

➢ **Requirements TMM- GEN-6:** Check arrays dimensions and avoid generic oversized arrays, especially because it might lead to memory problems (access to array beyond its dimension).

16.2 Esatan to Thermisol requirements

This section lists the actions required when importing an Esatan model with Thermisol.

- ➢ **Requirements TMM-IMP-0:** The mathematical model shall be compliant with the generic TMM requirements listed above.

- ➢ **Requirements TMM-IMP-1:** The mathematical model shall successfully run with Esatan and referenced results shall be provided in a suitable format for validation purpose.

- ➢ **Requirements TMM-IMP-2:** After executing the model with Thermisol, all warnings and errors raised by the Thermisol pre-processor shall be checked. If modifications are required, it is recommended to get new reference results obtained with the modified model.

- ➢ **Requirements TMM-IMP-3:** Thermisol results shall be validated against the Esatan reference results.

16.3 Thermisol to Esatan requirements

This section lists the actions required when exporting a Thermisol model to Esatan.

- ➢ **Requirements TMM-EXP-0:** The mathematical model shall be compliant with the generic TMM requirements listed above.

- ➢ **Requirements TMM-EXP-1:** Thermisol converter shall be used to make sure that the model is correctly formatted for being executed with Esatan. The converted model shall run with Thermisol and give the same results as the original model (differences due to the use of the $VARIABLES1 block shall be minimized by reducing the time-step if need be).

- ➢ **Requirements TMM-EXP-2:** The (converted) mathematical model shall successfully run with Thermisol and referenced results shall be provided in a suitable format for validation purpose.

- ➢ **Requirements TMM-EXP-3:** After executing the model with Esatan, all warnings and errors raised by the Esatan syntax checker shall be checked. If modifications are required, it is recommended to get new reference results obtained with the modified model.

- ➢ **Requirements TMM-EXP-3:** Esatan results shall be validated against the Thermisol reference results.

Results Conversion

This section is empty for the moment.