Delphin 6 Material File Specification, Version 6.0

Technical Report

Stefan Vogelsang, Heiko Fechner, Andreas Nicolai

Institut für Bauklimatik
Technische Universität Dresden
D-01062 Dresden, Germany
stefan.vogelsang@tu-dresden.de

Abstract

This paper describes the format of material data files that hold parameters needed by thermal and hygrothermal simulation tools such as *Delphin, Hajawee (Dynamic Room Model)* and *Nandrad*. The Material Data Files are containers for storing parameters and functions for heat and moisture transport and storage models. The article also discusses the application programming interface of the *Material* library that can be used to read/write *material* data files conveniently and efficiently.

Contents

1 Introduction

Hygrothermal and thermal transport models require a number of material properties and parameters. Depending on the model setup, different material functions may be needed. The Material File Format is a container holding several typically used material properties and functions for different simulation models. It serves as standard container and file-based database format for the material database of the Institute of Building Climatology (Institut für Bauklimatik, IBK) of the Technische Universität Dresden, Germany.

The file format is an ASCII Format and is designed such that it can be created and edited with plain text editor and can be manipulated easily via scripts. This article describes the Material File Format Version 6 with the standard file extension `m6` that supersedes previous material file format Version 5 (extension `mat`). Since both formats are very similar, the article will point out renamings and restructuring in order to facilitate manual conversion of material files from Version 5 to Version 6 format[1].

Minimalistic Example

Providing a brief overview of the material file format, we include a minimalistic example of such a material data file. Figure 1 shows the content of a small material data file. This file can be used, for example, in the thermal transport models *Hajawee* and *Nandrad*.

Listing 1: Minimalistic Material Data File Example with all Parameters for a Typical Thermal Transport Model

```
D6MARLZ! 006.000

[IDENTIFICATION]
    NAME                     = de: Altbauziegel Bolonga|en: Old Building Brick Bologna

[STORAGE_BASE_PARAMETERS]
    RHO                      = 1758.96 kg/m3
    CE                       = 1092.18 J/kgK

[TRANSPORT_BASE_PARAMETERS]
    LAMBDA                   = 0.6235 W/mK
```

This example material file begins with the magic header (see §2) and contains three sections. The identification section (see §5.3) simply defines the name of the material, using a multi-language string definition. The section with the storage base parameters (see §5.4) defines bulk density and specific heat capacity. Finally, the transport base parameters section (see §5.5) defines the thermal conductivity. This material file defines all properties needed for a typical thermal transport model.

2 Magic Headers

All material files begin with the same header structure stored in the first 16 bytes of each file. The first 4 bytes encode the identification of a file. This information is used to automatically select the correct parser for each material file format. Currently, only ASCII format is supported and a valid Version 6 Material file is expected to begin with the magic number `0x414D3644.` The following 4 bytes must be equal to the number `0x215A4C52.` This allows disambiguation with other magic file headers and thus guarantees a unique identification in the UNIX world. The first 16 Byte of each Material file are in ASCII representation:

```
D6MARLZ!
```

Version Number Encoding

The next 8 bytes encode the file format version number consisting of a major and minor number. All minor version changes are downwards compatible. No content sections in the files will be removed or altered when a new minor version is released. Consequently, applications supporting older versions of file formats can still read newer file formats. For example, in minor version changes additional meta-information may be added to the file that can simply be ignored.

Major version changes indicate significant changes in the file format and result in incompatible files. A major revision number change might require updates for older distributions of simulation software using the material data files.

The maximum major and minor version number is limited to 255, corresponding to an 8 bit number. This limit is introduced to ensure that the ASCII representation of a version number has at maximum 3 digits. Since this

[1] Version 5 was used in the hygrothermal simulation codes Delphin 5.0 to 5.6, Version 6 is used for Delphin Versions 5.8 and newer. The Version 6 file format is also used in the simulation models Hajawee and Nandrad.

document specifies already version 6.0 of the file format, $249 * 256 = 63744$ version numbers are left within this format number definition. The ASCII Version number must have exactly 8 characters (64 bit), as in the example below.

Listing 2: Version Number Encoding Example

```
␣006.015
```

The version string begins with a single space character, followed by the version numbers. Major and minor version numbers are separated by a . (dot) character. Both version numbers must have exactly 3 digits, using the 0 as fill character.

3 File Name Conventions

While not part of the file format specification itself, a naming convention of file names is meaningful to organize multiple material data files and facility file-based data storage. Each material data file shall be formed using the following pattern:

```
<materialName>_<uniqueID>.m6
```

`<materialName>` corresponds to a descriptive name of the material in the file and is primarily used to organize and identify material files when working directly with filenames. Within material names only upper and lower case letters are valid. No non printable, or special characters, or umlauts are allowed.

`<uniqueID>` is a unique ID for identification in a database. When managing material files in a database view, as done in the IBK simulation codes, each material file must have an unique ID.

It is recommended to use new IDs when altering material data files to keep track of a history of changes. Modified material data files shall refer to the original material file ID by storing appropriate meta-information (see §5.3).

4 Format Description

The ASCII material data files are encoded in *UTF8*, utilizing a line-based storage protocol. Each line may contain either a section header definition, a keyword definition, a comment, or a raw data line. A comment starts with a # (hash) character and ends with the line end.

The data file is structured into sections. Indentation is optional and typically used to increase readability of the file as in the following example.

Listing 3: Section Header Example

```
[<SECTION_HEADER>]

    <KEYWORD> = <VALUE>
    <KEYWORD> = <VALUE>
    <KEYWORD> = <VALUE>
    <KEYWORD> = <VALUE>
    ...


[<SECTION_HEADER>]
    <KEYWORD> = <VALUE>
    <KEYWORD> = <VALUE>
    ...
```

Section headers are defined by unique identifier strings enclosed in square brackets. Per line only one header definition is allowed. Each section must be defined only once per ASCII material file. Lines containing a section header must not contain other data. Version 5 sub section headers are obsolete and not supported in Version 6.

Within sections data is stored in keyword-value format:

```
<KEYWORD> = <VALUE>
```

wherein keywords and values are separated by the first = (equal sign) character that appears in the line. Whitespace characters (tab, space) around keyword and value are ignored/trimmed. A keyword definition ends with a line break. If no value is present for a keyword, the complete data line *must be omitted*. Keyword names are defined with respect to the section they appear in. They are generally not unique within the file so that the section context needs to be taken into account. Keywords are capitalized strings without special characters.

Typical data types for values are strings/identifiers, internationalized strings, and parameters.

4.1 String/Identifier Value Type

A string/identifier value type is any *UTF8*-encoded string.

4.2 Internationalized/Translated String Value Type

To support translations into other languages an encoded table syntax is used for such a string:

```
<language-code>:<text>|<language-code>:<text>|...
```

Each translation starts with a case-insensitive two-letter-language-code derived from ISO-639 standard followed by a : (colon) character. The text for that language follows. If no language code is specified the complete string is used for all languages and no further translations are searched for this keyword. Different language strings are separated by a | character. This specification refers to this string format as *multi-language string*.

Listing 4: Multi-Language String Example

```
de:Ziegel|en:Brick|fr:tuile
```

4.3 Parameter Value Type

Parameters are stored using a numeric value followed by a unit string.

```
<KEYWORD> = <NUMBER> <UNIT>
```

Numbers are double precision numbers in general or scientific notation in English number format. The latter uses the format `1.23e45` for powers of 10. Thousands separators must not be used.

Listing 5: Example Parameter

```
RHO = 2130 kg/m3
```

4.4 Special Data Lines and End-of-File Convention

Raw-data lines are space separated series of data values ending with a line break. Semantics of raw-data depend on the section they are belonging to. Please refer to the following sections for precise definitions of different raw data line formats.

A material file closes with an empty line containing only a line break before the end-of-file marker.

4.5 Special Keyword `FUNCTION`

The `FUNCTION` keyword indicates storage of data tables with x-y value pairs that describe a function y(x). A function definition follows the syntax:

```
FUNCTION = <FUNCTION_IDENTIFIER>
<raw data x-value vector>
<raw data y-value vector>
```

A line with a FUNCTION keyword must be followed by two raw-data lines. These two lines hold the x and y value vectors of the data table. Numbers in each vector are whitespace separated. Double precision English number format is used for the numeric values, without thousands separators. Both raw vectors are coupled by index, thus both vectors need to have the same number of numeric values. The data tables are supposed to be linearly interpolated. Therefore, x-values must be increasing strictly monotonic. The physical meaning and units are uniquely defined through the function identifier and the section context.

Listing 6: Example Keyword FUNCTION

```
FUNCTION = Theta_l(RH)_de
  0  0.3    0.6    0.73    0.8   0.93  1
  0  0.005  0.008  0.0092  0.06  0.25  0.33
```

5 Material File Sections

5.1 Section Keywords

Material data file content is structured into several sections:

```
[IDENTIFICATION]
[STORAGE_BASE_PARAMETERS]
[TRANSPORT_BASE_PARAMETERS]
[MECHANICAL_BASE_PARAMETERS]
[MOISTURE_STORAGE]
[MOISTURE_TRANSPORT]
[HEAT_TRANSPORT]
[AIR_TRANSPORT]
```

The material data file contains a parameter block for each section definition, named by a unique section keyword. If anisotropic material properties are to be stored, the following transport-related sections may be additionally present:

```
[MOISTURE_TRANSPORT_V]
[MOISTURE_TRANSPORT_W]
[HEAT_TRANSPORT_V]
[HEAT_TRANSPORT_W]
[AIR_TRANSPORT_V]
[AIR_TRANSPORT_W]
```

The appendix `_V` and `_W` correspond to transport directions V and W, a transport direction U is described by the aforementioned standard transport property sections `[MOISTURE_TRANSPORT]`, `[HEAT_TRANSPORT]` and `[AIR_TRANSPORT]`. A description of an abstract material direction to an real world material direction name can be specified in the `IDENTIFICATION` section, please note § 5.3.

5.2 Completeness of Material Data

Generally, the material data file is a container format for a variety of parameters and meta information. This format description only lists the possible entries currently known and used by corresponding software packages. It depends on a models simulation code to define which parameters are needed by the modelled equations. Therefore, this document describes physical quantities alongside their mathematical symbol. The individual model descriptions use these symbols in model equations and define own requirements on the completeness of the material data. Consequently, all sections and their parameters are optional.

5.3 Section `[IDENTIFICATION]`

The identification section contains meta-information that describe a material parametrised in this file. This information is mainly used to distinguish similar materials as well as for filtering and grouping of material entries in database views.

Keyword	Description/Value Format
NAME	Name of a material is a multi-language-string following the pattern: <generalMaterialName>["[<producerMaterialName>]"].
PRODUCER	Full name of the material producer as multi-language-string.
PRODUCT_ID	The producer's product identification as multi-language-string.
INVESTIGATOR	Contact to the investigating laboratory or contact person (test certificate) as string.
LABORATORY	The institute that provides or measured the material data as multi-language-string.
DATE	The date and time the output file was written. Stored as a string utilizing Internet RFCs 2822 format. For example: "Fri, 09 Sep 2005 13:51:39 -700"
SAMPLING_DATE	The date and time the samples were acquired. Stored as a string utilizing Internet RFCs 2822 format. For example: "Fri, 09 Sep 2005 13:51:39 -700"
PRODUCTION_DATE	The date and time the samples were produced. Stored as a string utilizing Internet RFCs 2822 format. For example: "Fri, 09 Sep 2005 13:51:39 -700"
SAMPLE_ID	The sample ID string used by the IBK laboratory to identify the sample.
SAMPLING_METHOD	The way the samples were produced for this material file is identified via one keyword according to the following table: S_CAST_CUT_PRODUCER — cast and cut by producer S_CAST_PRODUCER — cast by producer S_CAST_CUT_IBK — cast and cut by IBK S_CAST_IBK — cast by IBK
COUNTRY	The country in which the material was acquired under the above mentioned material name stored as string. Please note a materials name or parameters may change in different countries.
COLOUR	The colour assigned to the material. Starting with "#" followed by the RRGGBB colour code in hexadecimal numbers. RR are the two red hex-digits, GG are the two green hex-digits, and BB are the two blue hex-digits.
FLAGS	All combinations of flag keywords specified in Table 3. All assigned flags are separated by a space character.
USE_INSTEAD	If specified, another material exists that has updated/improved parametrization and should be used instead of this material data set. Value should be a number: 0 — deprecated material, no replacement 1...n — deprecated material, replaced by ID given here
COPYRIGHT	The copyrights of the material data owner as multi-language string.
CATEGORY	The material's classification as whitespace separated list of category keywords. Per material three categories at max. can be assigned. The categories are ordered by importance. Categories NATURAL_MATERIALS and MISC must never be the first category. Please note Table 2.
COMMENTS	Additional information a material data provider wants to add (e.g. "en: material was created with material creator version 1.3.") stored as multi-language-string.
U_DIRECTION	Display name of u - direction according to an anisotropic material as multi-language string. For isotropic materials this is the default name.
V_DIRECTION	Display name of v - direction according to an anisotropic two - or three - dimensional material structure as multi-language string.
W_DIRECTION	Display name of w - direction according to an anisotropic three-dimensional material structure. Left blank for one - or two - dimensional materials as multi-language string.
DB_TYPE	Classifies a materials affiliation to IBK provided databases. In case of special software versions (e.g. for material producers) a selection of the shown materials is necessary. (e.g. some material data is confidential). In this case the database type gives information if a specific material should be included in this database or not. Third-party material data providers should set 0 here. Please see Table 4 for the correct unsigned integer number to be selected here.
SIGNATURE_SOURCE	A link to the signature source location to verify this material.
DATA_SHEET_SOURCE	A link to the producers data sheet.

Table 1: Keywords in Identification Section

Obsolete keywords in the identification section are SOURCE (previously used to describe origin of material parameters), MASTER_FORMAT1, MASTER_FORMAT2, and MASTER_FORMAT3 (previously used to categorize materials).

The flag HYGROSCOPIC_RANGE_ONLY is deprecated (see Table 3 for a list of possible flags).

Keyword	Description/Value
COATING	Coating
PLASTER	Plaster and Mortar
BRICK	Building Brick
NATURAL_STONES	Natural Stones
CONCRETE	Concrete Containing Materials
INSULATION	Insulation Materials
BUILDING_BOARDS	Building Boards
TIMBER	Timber
NATURAL_MATERIALS	Natural Materials
SOIL	Soil
CLADDING	Cladding Panels and Ceramic Tiles
FOILS	Foils and Waterproofing Products
MISC	Miscellaneous

Table 2: Material Categories

Keyword	Description
AIR	Air material (air gap, etc.)
AIR_TIGHT	Material is air-tight (no air flow allowed)
VAPOUR_TIGHT	Material is vapour-tight (no vapour diffusion/convection allowed)
WATER_TIGHT	Material is water-tight (no capillary liquid suction allowed)
WOOD	Material is a wood
WOOD_BASED	Material is made from wood or contains wood
FOILS	Foil or coating (no storage capacity defined)

Table 3: Material Flags

Number	Description	Removed Keyword
0	Default Database	Standard
1-16000	Reserved Space (IBK).	-

Table 4: Database Types

Listing 7: Example for Typical Content in an Identification Section

```
[IDENTIFICATION]
    NAME             = en: Oak wood |de: Eichenholz
    FLAGS            = WOOD
    SOURCE           = TUD, MaterialGenerator v1.0.3.9
    CATEGORY         = TIMBER NATURAL_MATERIALS
    U_DIRECTION      = en: Radial |de: Radial
    V_DIRECTION      = en: Tangential |de: Tangential
    W_DIRECTION      = en: Longitudinal |de: Longitudinal
    DB_TYPE          = 0
    DATE             = Thu, 22 Nov 2011 15:30:39 +100
    COUNTRY          = en: Switzerland |de: Schweiz
    COLOUR           = #409020
```

5.4 Section [STORAGE_BASE_PARAMETERS]

Parameters related to heat and moisture storage are stored in this section. The use of these parameters in models or purely for identification and comparison purposes is determined by the model using the data. The Digits column in Table 5 and the following parameter tables holds recommendations for a minimum accuracy (number of significant digits) to be used for numbers.

Keyword	Description/Value Format	Unit	Digits	Symbol	Obsolete Name
RHO	Density of dry material including pores (bulk density).	kg/m^3	2	ρ	
CE	Specific heat capacity at constant pressure. Energy needed to increase the temperature of 1 kg of dry material by 1 K.	J/kgK	1	c_p	
THETA_POR	Open porosity, does not include closed pores. The open porosity must be greater or equal then the effective saturation moisture content.	m^3/m^3	4	θ_{por}	OPOR
THETA_EFF	Effective saturation moisture content. The effective saturation moisture content is a long term (maximum) saturation. It must be greater or equal than the capillary saturation moisture content.	m^3/m^3	4	θ_{eff}	OEFF
THETA_CAP	Capillary saturation moisture content. Mean moisture content of a sample obtained in the water uptake experiment at the end of the first water uptake period. The capillary saturation moisture content is a short term saturation. It is determined when the water front reaches the top of the specimen.	m^3/m^3	4	θ_{cap}	OCAP
THETA_80	Hygroscopic moisture content at RH=80% obtained in an hygroscopic adsorption experiment. It should match the (ad)sorption isotherm at RH=80%.	m^3/m^3	4	θ_{80}	O80
THETA_LIM_HYG	Limitation hygroscopic moisture content for those materials that must not get wet. May be used as indicator for materials that shall be subjected to hygroscopic moisture loads only.	m^3/m^3	4	$\theta_{lim,hyg}$	OLIMHYG

Table 5: Keywords in Storage Base Parameters Section

Listing 8: Example Section with Storage Base Parameters

```
[STORAGE_BASE_PARAMETERS]
    RHO          = 588.257 kg/m3
    CE           = 1363.84 J/kgK
    THETA_POR    = 0.629973 m3/m3
    THETA_EFF    = 0.618 m3/m3
    THETA_CAP    = 0.485 m3/m3
    THETA_80     = 0.073666 m3/m3
```

5.5 Section [TRANSPORT_BASE_PARAMETERS]

For each anisotropic direction u, v, and w a transport based parameter can be defined. The corresponding keywords follow the pattern:

```
<keyword>[_<direction>]
```

where direction can be U, V and/or W. For instance `LAMDA_U` and `LAMDA_V` may be defined for an anisotropic two-dimensional material. If the separator "`_`" and the according direction is omitted, the use of "`_U`" is assumed. If an anisotropic material is used for isotropic calculations the "`_U`"-values are considered to be the isotropic material parameters.

Keyword	Description/Value Format	Unit	Digits	Symbol
LAMBDA	Thermal conductivity for the dry material.	W/mK	4	λ_{dry}
LAMBDA_DESIGN	Design value for thermal conductivity for material. These values are given from material producer and not used for calculation.	W/mK	4	λ_{design}
AW	Water uptake coefficient.	$kg/m^2\sqrt{s}$	4	A_w
MEW	Water vapour diffusion resistance factor (dry cup).	$-$	1	μ
KLEFF	Liquid water conductivity at effective saturation.	s	2	$K_{\ell,eff}$
DLEFF	Liquid water diffusivity at effective saturation.	s	2	$D_{\ell,eff}$
KG	Air permeability of the dry material.	s	4	$K_{g,dry}$
SD	Equivalent air layer thickness. Relates the vapour resistance of a material (foils and coatings only) to the vapour resistance of air.	m	1	s_d

Table 6: Keywords in Transport Base Parameters Section

A typical base parameter section of a material file without direction indicator (i.e. isotropic material) looks like:

```
[TRANSPORT_BASE_PARAMETERS]
    LAMBDA       = 0.1557 W/mK
    AW           = 0.0088 kg/m2s05
    MEW          = 648.02 -
    KLEFF        = 2.0578e-011 s
```

Alternatively, _U could be appended to all properties, providing the same information:

```
[TRANSPORT_BASE_PARAMETERS]
    LAMBDA_U     = 0.1557 W/mK
    AW_U         = 0.0088 kg/m2s05
    MEW_U        = 648.02 -
    KLEFF_U      = 2.0578e-011 s
```

An anisotropic material may have additional V and W direction parameters:

```
[TRANSPORT_BASE_PARAMETERS]
    LAMBDA_U     = 0.1557 W/mK
    LAMBDA_V     = 0.1863 W/mK
    AW_U         = 0.0088 kg/m2s05
    AW_V         = 0.0065 kg/m2s05
    MEW_U        = 648.02 -
    MEW_V        = 729.05 -
    KLEFF_U      = 2.0578e-011 s
    KLEFF_V      = 1.7220e-011 s
```

5.6 Section [MECHANICAL_BASE_PARAMETERS]

For each anisotropic direction u, v, and w a transport based parameter can be defined. The corresponding keywords follow the pattern:

```
<keyword>[_<direction>]
```

where direction can be U, V and/or W. For instance `ALPHA_U` and `ALPHA_V` may be defined for an anisotropic two-dimensional material. If the separator "`_`" and the according direction is omitted, the use of "`_U`" is assumed. If an anisotropic material is used for isotropic calculations the "`_U`"-values are considered to be the isotropic material parameters.

Keyword	Description/Value	Unit	Digits	Symbol	Obsolete Name
ALPHA	Thermal expansion coefficient. The thermal expansion is calculated by the product of thermal expansion coefficient and temperature difference (in K). The thermal expansion coefficient is used for conversion of the temperature field into an expansion field.	$\frac{mm}{mK}$	1	α	
BETA_THETA_L	The hygric expansion is calculated by the product of hygric expansion coefficient and moisture content difference in (m3/m3). The hygric expansion coefficient is used for conversion of the water content field into an expansion field.	$\frac{m^3m}{m^3m}$	1	$\beta(\theta_\ell)$	BETA_O_L
BETA_THETA_L2	Fibre saturation point.	m^3/m^3	4	$\beta_2(\theta_\ell)$	

Table 7: Keywords in Mechanical Base Parameters Section

Listing 9: Example for a Mechanical Base Parameters Section

```
[MECHANICAL_BASE_PARAMETERS]
    ALPHA         = 0.1   mm/mK
    BETA_THETA_L  = 0.2   m3m/m3m
    BETA_THETA_L2 = 0.0012 m3/m3
```

5.7 Section [MOISTURE_STORAGE]

The moisture retention characteristics describes the moisture content in a material versus a moisture potential, which can be capillary pressure p_c or relative humidity φ. This function can be specified either as moisture retention function $\theta_\ell(p_C)$ where $p_C = \log_{10}(-p_c)$ or sorption isotherm $\theta_l(\varphi)$ using data tables (see §4.5). The moisture retention function describes the over-hygroscopic moisture range in much more detail and is therefore recommended.

To enable hysteretic moisture storage behaviour adsorption and desorption moisture retention curves are distinguished by suffixes `_de` and `_ad`. To parametrise moisture storage functions the `FUNCTION` keyword is used. Table 8 lists all function identifiers and their respective properties.

Function	Description/Value Format	Unit	Digits	Symbol	Obsolete Name
Theta_l(pC)_de	Moisture retention curve - desorption	m^3/m^3	4	$\theta_\ell\,(p_C)$	Ol(pC)
Theta_l(pC)_ad	Moisture retention curve - adsorption	m^3/m^3	4	$\theta_\ell\,(p_C)_{ad}$	Ol(pC)_ad
pC(Theta_l)_de	Reversed moisture retention curve - desorption	$log_{10}(Pa)$	4	$p_C\,(\theta_\ell)$	pC(Ol)
pC(Theta_l)_ad	Reversed moisture retention curve - adsorption	$log_{10}(Pa)$	4	$p_C\,(\theta_\ell)_{ad}$	pC(Ol)_ad
Theta_l(RH)_de	Sorption isotherm - desorption	m^3/m^3	4	$\theta_\ell\,(\varphi)$	Ol(RH)
Theta_l(RH)_ad	Sorption isotherm - adsorption	m^3/m^3	4	$\theta_\ell\,(\varphi)_{ad}$	Ol(RH)_ad

Table 8: Function Identifiers in the Moisture Storage Section

Only one representation for desorption and adsorption moisture storage functions is allowed in the parametrization. That means, if both $\theta_\ell\,(p_C)$ and $\theta_\ell\,(\varphi)$ data is available, only one of the two data sets may be stored in the material file[2]. Generally, $\theta_\ell\,(\varphi)$ should be preferred. When storing non-hysteretic moisture storage parameters, the `_de` versions of the material functions *must be used*.

[2]Since both functions are interrelated through the Kelvin equation storing both functions would add redundant information to the material data file. This also bears the potential risk of storing mismatching data in the file and shall therefore be avoided.

The following value ranges are recommended:

- $p_C \geq 0$
- $0 \leq \theta_\ell \leq \theta_{\ell,eff}$
- $0 \leq \varphi \leq 1$

Note that the x-values of each function should be specified with enough accuracy to ensure strict monotonic increase of x-values.

Listing 10: Example for a Moisture Storage Section

```
[MOISTURE_STORAGE]
    FUNCTION = Theta_l(pC)_de
     0                 0.05              0.1               0.15              ...
     0.900184          0.900158664343    0.900129290173    0.900095406956    ...

    FUNCTION = pC(Theta_l)_de
     0                 0.000131232405977  0.000153209554415  0.000175186702852  ...
     12                9.375              8.65184783936      8.62365722656      ...
```

5.8 Section [MOISTURE_TRANSPORT]

As described in §5.1, storage of anisotropic material properties is possible through adding suffixes _U, _V and/or _W to the transport parameter sections, for example: `MOISTURE_TRANSPORT_U` and `MOISTURE_TRANSPORT_V`. Within each section the keywords are named the same, therefore the documentation below is applicable for all transport directions.

The moisture transport section contains function definitions to describe capillary liquid moisture transport and water vapour diffusion. With respect to capillary transport, two models can be alternatively used. The liquid conductivity model (1) describes the liquid moisture flux density j^w in kg/m^3 driven by a liquid pressure gradient (gravity term omitted here). The liquid conductivity $K_\ell(\theta_\ell)$ in s is the material function that needs to be parametrized for this model.

$$j^w = -K_\ell \nabla p_\ell \tag{1}$$

The moisture diffusivity model (2) describes liquid moisture flux density j^w driven by a moisture content gradient. The moisture diffusivity $D_\ell(\theta_\ell)$ in m^2/s needs to be parametrized.

$$j^w = -\varrho_w D_\ell \nabla \theta_\ell \tag{2}$$

Within the moisture transport section, either K_ℓ or D_ℓ can be specified, but not both[3].

With respect to water vapour diffusion, the driving force for vapour diffusion is in all cases a gradient of vapour pressure p_v. The vapour diffusion flux density j^v can be described in three alternative formulations:

$$j^v = -K_v \nabla p_v \tag{3}$$

$$= -\frac{D_v}{R_v T} \nabla p_v \tag{4}$$

$$= -\frac{D_{v,air}}{\mu R_v T} \nabla p_v \tag{5}$$

Equation (3) uses a vapour permeability/conductivity K_v in $[s]$ as transport function. Equation (4) shows the relation of the vapour diffusivity D_v to K_v. In a material file D_v cannot be parametrised directly. Instead, D_v values should be converted to K_v values using a reference temperature of 293.15 K:

$$K_v = D_v / (R_v \, 293.15\,[\mathrm{K}])$$

where $R_v = 462$ J/kgK is the gas constant for water vapour. Finally, equation (5) uses the water vapour diffusion resistance factor μ to relate the diffusivity of water vapour in air $D_{v,air}$, equation (6), to that of the material.

[3] Redundant information, in particular redundant material functions within the material file, should be avoided.

$$D_{v,air} \quad = \quad 0.083\frac{p_{ref}}{p}\left(\frac{T}{T_{ref}}\right)^{1.81} = 0.083\left[\frac{m^2}{s}\right]\frac{101323\ [Pa]}{p}\left(\frac{T}{273.15\ [K]}\right)^{1.81} \approx 26.1\cdot 10^{-6}\left[\frac{m^2}{s}\right] \qquad (6)$$

Within the moisture transport section, either $K_v(\theta_\ell)$ or $\mu(\theta_\ell)$ can be parametrized, but not both. If neither of the two functions is present, equation (5) can still be used with the constant μ-value defined in the transport base parameter section (see §5.5).

Table 9 describes function identifiers for moisture transport in material files, and their corresponding properties. Note, due to large non-linearity of these material functions, they are specified via their logarithmic values.

Function	Description/Value Format	Unit	Digits	Symbol	Obsolete Name
`lgKl(Theta_l)`	Capillary conductivity (logarithmic) depending on moisture content.	$log_{10}(s)$	4	$\lg K_\ell(\theta_\ell)$	`lgKl(Ol)`
`lgDl(Theta_l)`	Liquid water diffusivity (logarithmic) depending on moisture content.	$log_{10}(m^2/s)$	4	$\lg D_\ell(\theta_\ell)$	`lgDl(Ol)`
`lgKv(Theta_l)`	Water vapour permeability (logarithmic) depending on moisture content.	$log_{10}(s)$	4	$\lg K_v(\theta_\ell)$	`lgKv(Ol)`
`mew(RH)`	Water vapour diffusion resistance factor depending on relative humidity. This function can be used for materials which are only valid in hygroscopic range (e.g. moisture dependent vapour retarder). In this case the material should have the `WATER_TIGHT` flag. If the `FOIL` flag is set, this μ - value corresponds to a foil thickness of 1mm.	–	4	$\mu(\varphi)$	`My(Phi)`

Table 9: Function Identifiers in Moisture Transport Section

Listing 11: Example for a Moisture Transport Section Storing Functions K_ℓ and K_v

```
[MOISTURE_TRANSPORT]
    FUNCTION   = lgKl(Theta_l)
      0.000200372          0.000416266       0.000422101       0.000427936       0.000433771   ...
      -20                  -19.8944          -19.8119          -19.7295          -19.5647      ...

    FUNCTION   = lgKv(Theta_l)
      0                0.00131658       0.00263317       0.00394975       0.00526634         ...
      -10.9975         -10.9978         -10.9991         -11.0006         -11.0022           ...
```

5.9 Section [HEAT_TRANSPORT]

As described in §5.1, storage of anisotropic material properties is possible through adding suffixes _U, _V and/or _W to the transport parameter sections, for example: `HEAT_TRANSPORT_U` and `HEAT_TRANSPORT_V`. Within each section the keywords are named the same, therefore the documentation below is applicable for all transport directions.

The heat transport section contains thermal transport functions. The heat flux density q, driven by temperature gradient, is computed using equation (7).

$$q = -\lambda\nabla T \qquad (7)$$

When a thermal conductivity λ is given for the dry material in the base transport parameters section (see §5.5), the moisture-dependent thermal conductivity can be computed with the following linear model (8).

$$\lambda(\theta_\ell) = \lambda_{dry} + \theta_\ell\, 0.56\,\frac{W}{mK} \qquad (8)$$

For more complex dependencies of thermal conductivity on either moisture content or temperature, the heat transport section can be used to store these functions. Table 10 lists function identifiers for moisture dependent thermal conductivity $\lambda(\theta_\ell)$ and temperature dependent thermal conductivity $\lambda(T)$ where T in [°C].

If only one of two functions is given, the thermal conductivity will be computed using the corresponding functional dependence. However, if both functions are given, equation (9) is used and $\lambda(\theta_\ell)$ and $\lambda(T)$ must be *normalised*, i.e. $\lambda_\theta(0) = 1$ and $\lambda_T(20°\mathrm{C}) = 1$.

$$\lambda(T, \theta_\ell) = \lambda_{dry}\, \lambda_T(T - 273.15\,\mathrm{K})\; \lambda_\theta(\theta_\ell) \tag{9}$$

Function	Description/Value Format	Unit	Digits	Symbol	Obsolete Name
`lambda(Theta_l)`	Thermal conductivity function depending on the moisture content.	W/mK	4	$\lambda_\theta(\theta_\ell)$	`lambda(Ol)`
`lambda(T)`	Thermal conductivity function depending on the temperature.	W/mK	4	$\lambda_T(T\,\mathrm{in}\,°\mathrm{C})$	

Table 10: Function Identifiers in Heat Transport Section

5.10 Section [AIR_TRANSPORT]

As described in §5.1, storage of anisotropic material properties is possible through adding suffixes _U, _V and/or _W to the transport parameter sections, for example: `MOISTURE_TRANSPORT_U` and `MOISTURE_TRANSPORT`_V. Within each section the keywords are named the same, therefore the documentation below is applicable for all transport directions.

Darcy flow models for air flow through porous materials are driven by gas pressure gradients. Equation (10) describes the dry air mass flux density j^a, with $K_g(\theta_\ell)$ as moisture-dependent material transport function (buoyancy terms are omitted). The transport base parameters section (see §(5.5)) defines already the constant $K_{g,dry}$ value. A moisture-dependent K_g value is parametrised as function in this material file section (see Table 11).

$$j^a = -K_g \nabla p_g \tag{10}$$

Function	Description/Value Format	Unit	Digits	Symbol	Deprecated Name
`Kg(Theta_l)`	Air permeability depending on the moisture content.	s	4	$K_g(\theta_\ell)$	`Kg(Ol)`

Table 11: Function Identifier in Materials Files Air Transport Section