QCOBJ Documentation

Release 1.2.0

Roberto Vidmar, Nicola Creati

Oct 17, 2018

CONTENTS

CHAPTER

ONE

FOREWORD

QCOBJ has been developed by two programmers working in the Aerial Operations group of the IRI Research Section at OGS - Istituto Nazionale di Oceanografia e di Geofisica Sperimentale.

Python is their favourite programming language since 2006.

The authors:

Roberto Vidmar, Nicola Creati

CHAPTER

TWO

WHAT IS QCOBJ?

Scientists often use configuration files (*cfg* files) to set the parameters and initial conditions for their computer programs or simulations. When these parameters are not limited to numbers or strings but represent physical quantities their unit of measure must be taken into account. Researchers are used to convert derived physical quantities by hand or with the help of some computer program but this operation slows down the process and is inherently error prone.

QCOBJ tries to give an answer to this problem integrating unit of measure and hence dimensionality into parameters. This approach ensures that programs using this package will always get numbers in the requested range and in the correct unit of measure independently of the units used in the configuration file.

CHAPTER

THREE

QCOBJ IN SHORT

- create/edit configuration files with the power of ConfigObj
- mix physical quantities at your pleasure: specify pressure parameter like
  - pressure = 300.0 Pa *or*
  - pressure = 0.03 N / cm**2 *or*
  - pressure = 2.842 kgf / ft**2

  and let QCOBJ handle the conversion for you.
- create validation files with physical quantities and valid range for all parameters.
- use the validation file to define the preferred physical quantities
- use `qcobj.CfgGui` to develop your own Qt based GUI to show/edit configuration files.

CHAPTER

FOUR

DEPENDENCIES

It relies on ConfigObj, Pint and PySide (or PyQt4 / PyQt5 , see `qcobj.qtCompat`) for the gui.

CHAPTER

FIVE

CREDITS

This package was possible thanks to Hernan E. Grecco <hernan.grecco@gmail.com> who released and mantains Pint and helped in the integration.

Note: The package depends on either PySide or PyQt (PyQt4 / PyQt5), and has ben tested with Python 3.4.3, 3.6.5, PySyde 1.2.4, PyQt4 4.8.1, PyQt5 5.10.1.

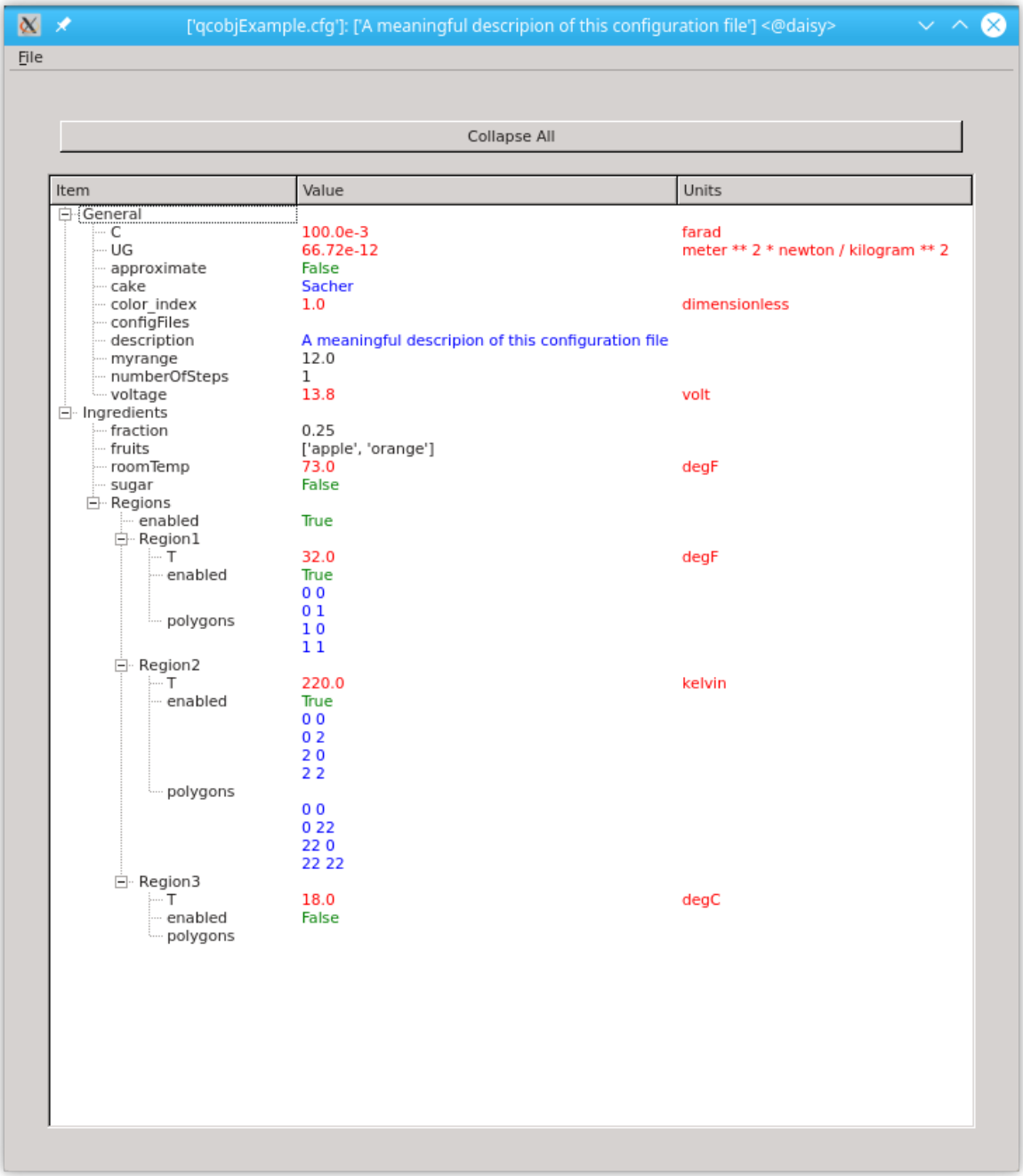
CHAPTER

SIX

MORE

6.1 CfgGui widget

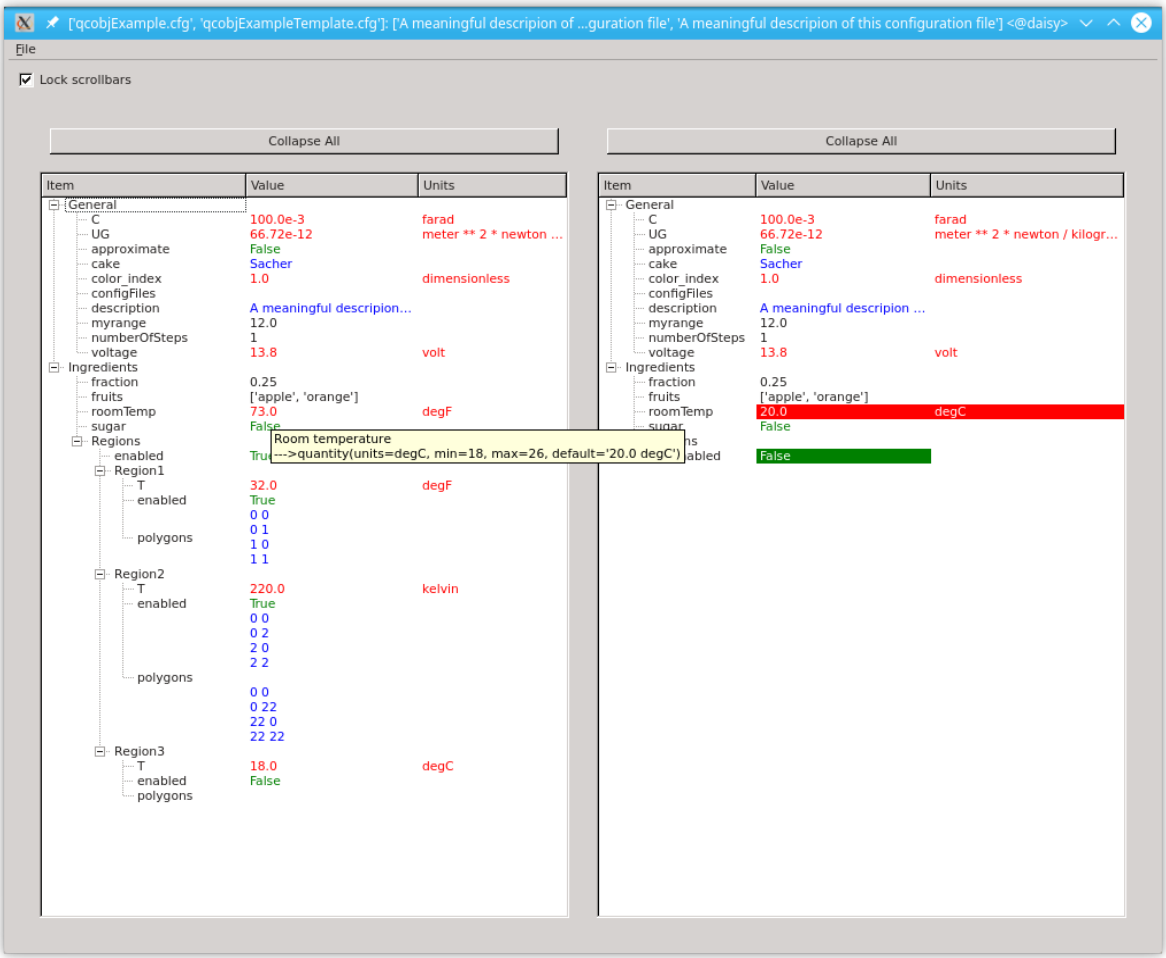
CfgGui widget showing qcobjExample



Values are coloured according to their type:

- Quantities: red
- Strings: blue
- Numbers and lists: black
- Boolean: green

The widget can also show two (or more!) configuration files at the same time highlighting the difference between them. This is what appears to the user runnning the script rungui

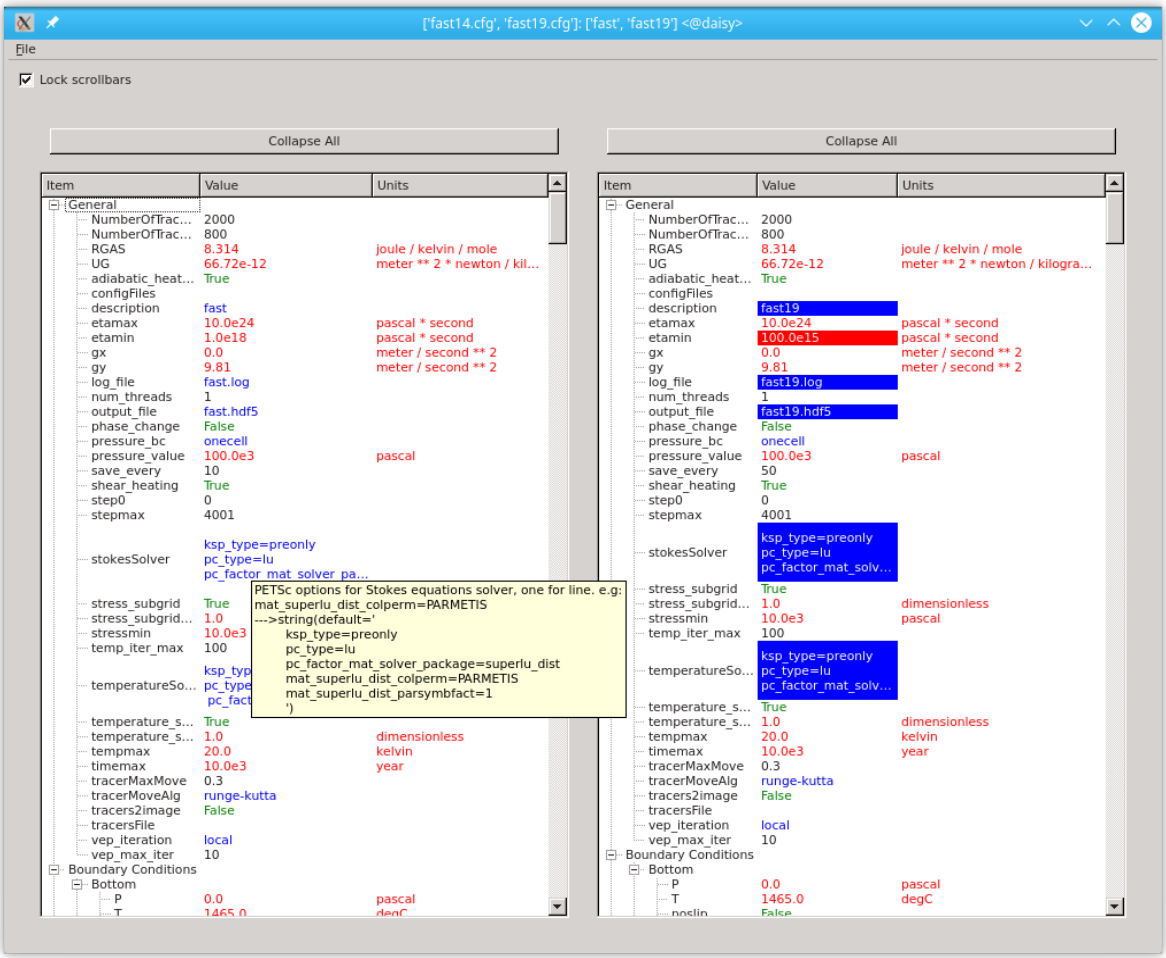


The two files that appear in this snapshot are qcobjExample and qcobjExampleTemplate.

Tooltips showing the valid quantities and range for every keyword appear when hoovering on a value.

6.2 A real example

This tool has been developed to compare large (more than one thousand lines) configuration files like fast14 against *fast19.cfg*



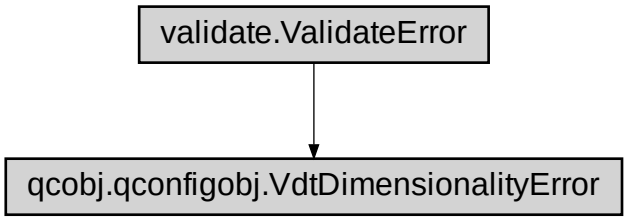
All values displayed can be modified and their value is validated against the configspec file that must be set when running the application.

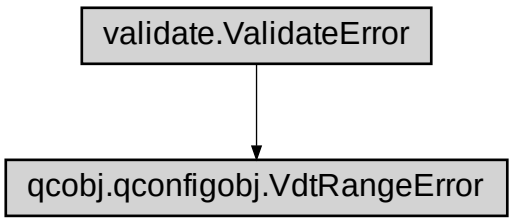
Modified configuration files can be saved using the *File* option in the application toolbar.

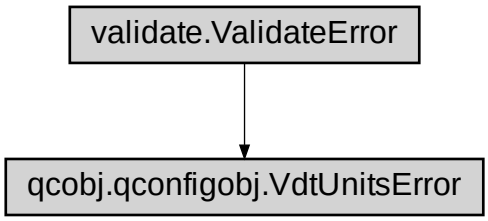
6.3 `qcobj` Package

6.3.1 `qconfigobj` Module

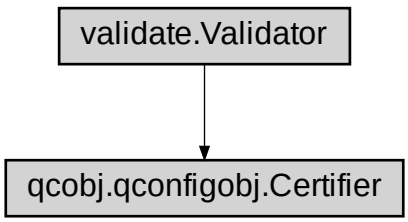
Validator errors classes:



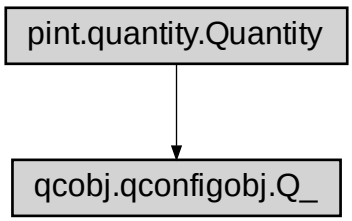


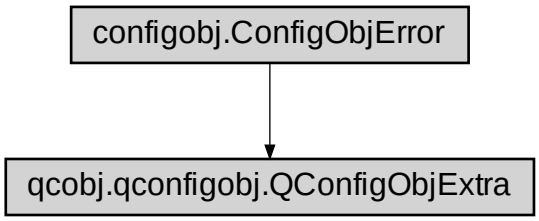


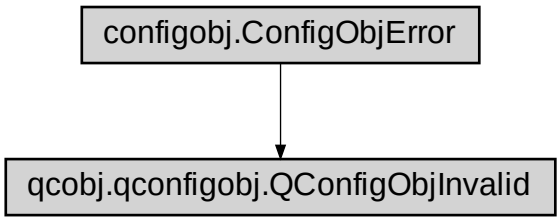
Validator class:



QConfigObj classes:







qcobj.qconfigobj.**eng_string**(*x*, *fmt='%s'*, *si=False*, *doround=None*)

Returns float/int value <x> formatted in a simplified engineering format using an exponent that is a multiple of 3.

Parameters

- **fmt** (*string*) – printf-style string used to format the value before the exponent.
- **si** (*boolean*) – if True, use SI suffix for exponent, e.g. k instead of e3, n instead of e-9 etc.
- **doround** (*boolean*) – if not None round the number to *doround* decimal digits

Returns: float/int value <x> formatted in a simplified engineering format using an exponent that is a multiple of 3.

```
E.g. with fmt='%.2f':
    1.23e-08 => 12.30e-9
```

```
        123 => 123.00
    1230.0 => 1.23e3
-1230000.0 => -1.23e6

and with si=True:
    1230.0 => 1.23k
-1230000.0 => -1.23M

and with doround=6
0.30000000000000004 => 0.3
```

qcobj.qconfigobj.**extract**(*string*, *start='('*, *stop=')'*, *firststop=True*)
Return substring between *start* and first/last *stop* characters

Parameters

- **string** (*string*) – the string to extract from
- **start** (*string*) – the left delimiter string
- **stop** (*string*) – the right delimiter string
- **firststop** (*bool*) – if True extract to the rightmost stop

Returns the extracted string

qcobj.qconfigobj.**isStringLike**(*s*)
Returns True if s acts "like" a string, i.e. is str or unicode.

Parameters **s** (*string*) – instance to inspect

Returns True if s acts like a string

qcobj.qconfigobj.**qLike**(*value*, *section*, *key*)
Return value converted to a quantity like *key* in *section*

Parameters

- **value** (*float or int*) – value to convert
- **section** (*qcobj.qconfigobj.QConfigObj*.Section') – instance
- **key** (*string*) – an existing key in *section*

Returns A *qcobj.qconfigobj.Q_* instance like the one at section[key] with magnitude *value*

qcobj.qconfigobj.**msec2cmyear**(*ms*)
Return m/s converted to cm/year Quantity

Parameters **ms** (*float*) – meters per second

Returns cm / year

qcobj.qconfigobj.**errors**(*cobj*, *ok*)
Return errors in a configuration file in terse format

Parameters

- **cobj** (*qcobj.qconfigobj.QConfigObj* instance) – instance
- **ok** (*dict*) – results dictionary returned by validate

Returns error messages string

qcobj.qconfigobj.**makeSpecEntry**(*key*, *spec*)
Accept new syntax suggested by Reviewer #1

Parameters

- **key** (*string*) – keyword

- **spec** (*dict*) – dictionary defining the quantity and valid range. Valid keywords:
  - comments: a comment string or a list of comment strings
  - units: a string with the units of measure
  - default: a default value (optional)
  - min: minimum value accepted (optional)
  - max: maximum value accepted (optional)

Returns configSpec string for *key*

qcobj.qconfigobj.**makeSpec**(*name*, *params*, *level=0*)
Create ConfigObj configspec string definition for section *name*

Parameters

- **name** (*string*) – name of the section we are building
- **params** (*Odict*) – ordered dict instance with the directives: The directives are (key, value) where value is a dictionary with the keywords:
  - comments: a comment string or a list of comment strings
  - units: a string with the units of measure
  - default: a default value (optional)
  - min: minimum value accepted (optional)
  - max: maximum value accepted (optional)
- **OLD** (*deprecated*) –
- **params** – ordered dict instance with the directives: The directives are (key, value) where value is a tuple of (comment, comment, ... comment, 'units, min, max', default). In case of int and float optional minimum or min and max values can be specified separated by ONE SINGLE blank char
- **level** (*int*) – indentation level

Returns configSpec string

qcobj.qconfigobj.**reindent**(*s*, *numSpaces=4*, *no_empty_lines=False*)
Return string s reindented by *numSpaces* spaces

Parameters

- **s** (*string*) – string to reindent
- **numSpaces** (*int*) – number of spaces to shift to the right
- **no_empty_lines** (*bool*) – if True remove empty lines

Returns reindented string

qcobj.qconfigobj.**setVal**(*section*, *key*, *value*)
Set *value* to section[key] converted to default units

Parameters

- **section** (*qcobj.qconfigobj.QConfigObj*.Section') – instance
- **key** (*string*) – valid key for *section*
- **value** (*float or int*) – value to set at section[key] converted to Quantity

qcobj.qconfigobj.**splitPolygons**(*s*)
Return a list of polygons from s. Separator is a blank line. Separation lines with blanks are digested as well.

Parameters **s** (*string*) – string defining polygon(s) separated by a blank line. One vertex per line

Returns list with polygons

qcobj.qconfigobj.**sumVal**(*section*, *key*, *value*)
Add *value* to section[key] converted to default units

Parameters

- **section** (*qcobj.qconfigobj.QConfigObj*.Section') – instance
- **key** (*string*) – valid key for *section*
- **value** (*float or int*) – value to add to section[key]

qcobj.qconfigobj.**toBaseUnits**(*q*)
Return magnitude of quantity *q* converted to base units * * Used for polygons * *

Parameters **q** (*qcobj.qconfigobj.Q_*) – instance

Returns magnitude of q in base units

qcobj.qconfigobj.**val**(*section*, *key*)
Return value of section[key] in units specified in configspec

Parameters

- **section** (*qcobj.qconfigobj.QConfigObj*.Section') – instance
- **key** (*string*) – valid key for *section*

Returns values in section[key] converted to units defined in configspec

qcobj.qconfigobj.**vval**(*d*, *k*)
Return magnitude in units specified in configspec for key *k' in qconfigobj section 'd* or simply magnitude if *d* is a dict instance

Parameters

- **d** (dict or *qcobj.qconfigobj.QConfigObj*.Section instance) – dict_like
- **k** (*string*) – key in d

Returns result of *qcobj.qconfigobj.val()* (d, k) or simply the magnitude of d[k]

class qcobj.qconfigobj.**Q_**
Bases: pint.quantity.Quantity

A Quantity class with user settable preferred units for representation

static **__new__**(**args*, ***kargs*)

__reduce__()

__str__()
Return *qcobj.qconfigobj.eng_string()* representation of magnitude rounded at 6 decimals with units

__repr__()
Return UnitRegistry Quantity representation

exception qcobj.qconfigobj.**QConfigObjInvalid**(*value*)
Bases: configobj.ConfigObjError

Invalid value error

__init__(*value*)

exception qcobj.qconfigobj.**QConfigObjExtra**(*value*)
Bases: configobj.ConfigObjError

Extra value / section error

__init__(*value*)

class qcobj.qconfigobj.**_QConfigObj**(**args*, ***kargs*)
Bases: configobj.ConfigObj

A Quantity aware ConfigObj class

__init__(**args*, ***kargs*)
Create a new instance. kargs are from ConfigObj

Keyword Arguments

- **infile** (*file instance*) – Input file (None)
- **configspec** (*list of strings*) – configspec (None)
- **encoding** (*string*) – encoding (None)
- **interpolation** (*bool*) – True
- **raise_errors** (*bool*) – False
- **list_values** (*bool*) – True
- **create_empty** (*bool*) – False
- **file_error** (*bool*) – False
- **stringify** (*bool*) – True,
- **indent_type** (*string*) – None
- **default_encoding** (*string*) – None
- **unrepr** (*bool*) – False
- **write_empty_values** (*bool*) – False
- **_inspec** (*bool*) – False
- **strict** (*bool*) – True
- **noextra** (*bool*) – True

_saveErrors()
Save errors in a configuration file in terse format

_extra()
Save extra values / sections in instance

_specAtPath(*path*)
Return configspec section at *path*

Parameters **path** (*list*) – list of section names

Returns configspec section at *path*

comment(*path*, *key*)
Return comment from configspec for *key* at *path*

Parameters

- **path** (*list*) – list of section names
- **key** (*string*) – valid key in section at *path*

Returns comment from configspec for *key* at *path*

validRange(*path*, *key*)
Return valid range for quantity *key* at *path*

Parameters

- **path** (*list*) – list of section names
- **key** (*string*) – valid key in section at *path*

Returns valid range for quantity *key* at *path*

configUnits(*section*, *key*)
Return units string for *key* in *section* or None

Parameters

- **section** (*qcobj.qconfigobj.QConfigObj*.Section) – instance
- **key** (*string*) – valid key in *section*

Returns Return units string for *key* in *section* or None

pretty()
Return pretty string representation for report attribute

Returns pretty string representation for report attribute

reference_quantity(*section*, *key*)
Return reference quantity for section[key]

Parameters

- **section** (*qcobj.qconfigobj.QConfigObj*.Section) – instance
- **key** (*string*) – valid key in *section*

Returns Return reference quantity for section[key]

Note: At present USED ONLY IN :class:GMOD2.boundaryCondition

---

val_to_default(*section*, *key*, *value*)
Set section[key] with value converted to default units (if any)

Parameters

- **section** (*qcobj.qconfigobj.QConfigObj*.Section) – instance
- **key** (*string*) – valid key in *section*
- **value** (*float or int*) – new value

Note: At present used only in addMag and setMag

---

write_to_string()
Return write content in a string

Returns Return write content in a string

class qcobj.qconfigobj.**QConfigObj**(**args*, ***kargs*)
Bases: *qcobj.qconfigobj._QConfigObj*

A Quantity aware ConfigObj class with CONFIGFILES_KEY support

__init__(**args*, ***kargs*)
Create a new instance. kargs are from ConfigObj

Keyword Arguments

- **infile** (*file instance*) – Input file (None)
- **configspec** (*list of strings*) – configspec (None)
- **encoding** (*string*) – encoding (None)
- **interpolation** (*bool*) – True
- **raise_errors** (*bool*) – False
- **list_values** (*bool*) – True

- **create_empty** (*bool*) – False
- **file_error** (*bool*) – False
- **stringify** (*bool*) – True,
- **indent_type** (*string*) – None
- **default_encoding** (*string*) – None
- **unrepr** (*bool*) – False
- **write_empty_values** (*bool*) – False
- **_inspec** (*bool*) – False
- **strict** (*bool*) – True
- **noextra** (*bool*) – True

exception qcobj.qconfigobj.**VdtUnitsError**(*value*)
Bases: validate.ValidateError

Missing *units* keyword in quantity type specifier

__init__(*value*)

exception qcobj.qconfigobj.**VdtDimensionalityError**(*dim1*, *dim2*)
Bases: validate.ValidateError

Dimensionality error class

__init__(*dim1*, *dim2*)

exception qcobj.qconfigobj.**VdtRangeError**(*value*, *vmin*, *vmax*, *units*)
Bases: validate.ValidateError

Range error class

__init__(*value*, *vmin*, *vmax*, *units*)

class qcobj.qconfigobj.**Certifier**(**args*, ***kargs*)
Bases: validate.Validator

A Validator for Quantities

See also:

Validator class

__init__(**args*, ***kargs*)

quantity_chek(*value*, **args*, ***kargs*)
Check if value has the right dimensions and is in the allowed range.

Quantity **MUST** be specified in configspec like: >>> quantity(units='Pa / s', min=0, max=100, default=50 Pa /s)

where:

- min and max *CAN* be positional arguments
- default value can be specified in any dimensionally correct unit after the first blank

Parameters **value** (*instance*) – the value we are checking

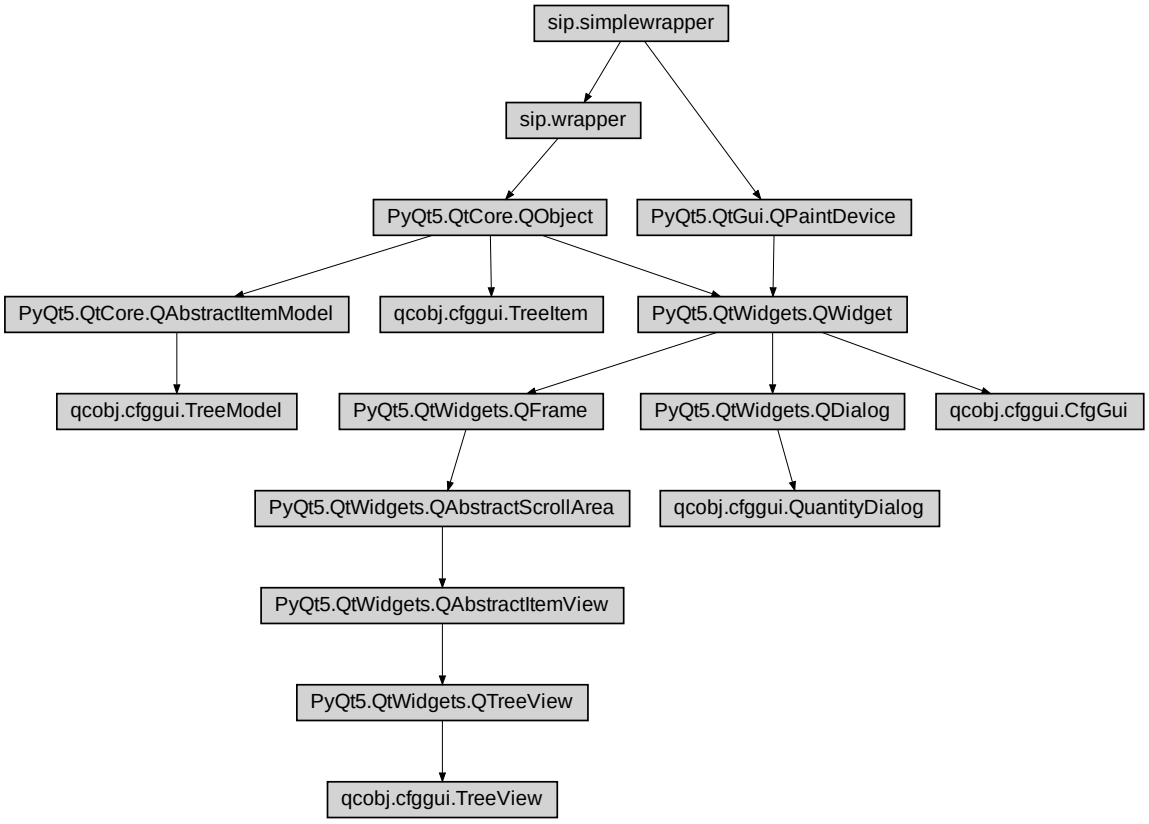
Returns validated quantity (or quantities)

Raises

- VdtUnitsError if no units are specified
- ValueError if *value* is not a quantity
- SyntaxError if quantity specification uses a wrong syntax

- *VdtDimensionalityError if value has the wrong physical* – dimension
- *VdtRangeError if value (converted to the units defined* – in configspec) is not in range [vmin, vmax]

6.3.2 `cfggui` Module



`qcobj.cfggui.`**`split_list`**(*L*, *n*, *stringify=True*)
Return a generator with the list *L* splitted in groups of *n* elements. If stringify evaluates as true, the groups of *n* elements are joined and terminated by

`qcobj.cfggui.`**`noBlanks`**(*withblanks*, *wordsPerLine=2*)
Remove blanks and format with *wordsPerLine* words per line

`qcobj.cfggui.`**`deBlank`**(*section*, *key*, *wordsPerLine=2*)
Remove blanks and format with *wordsPerLine* words per line every value with the key == 'polygon'

`qcobj.cfggui.`**`colorize`**(*s*, *color*)
Return an HTML colorized string for *s*

`qcobj.cfggui.`**`getPath`**(*index*)
Return section path at *index*

`qcobj.cfggui.`**`valueAtPath`**(*cobj*, *path*, *name*)
Return cobj value at *path* or raise RuntimeError

class qcobj.cfggui.**TreeItem**(*name=''*, *parent=None*, *data=None*)
Bases: PyQt5.QtCore.QObject

__init__(*name=''*, *parent=None*, *data=None*)

name()

appendChild(*item*)

child(*row*)

childCount()

columnCount()

data()

parent()

row()

setData(*value*, *validRange*, *column*)
Set node data with value converted to appropriate units as stated in validRange and return it or return None

class qcobj.cfggui.**TreeModel**(*parent*, *qcobj=None*)
Bases: PyQt5.QtCore.QAbstractItemModel

__init__(*parent*, *qcobj=None*)

columnCount(*parent*)

data(*index*, *role*)

flags(*index*)
Must be implemented

headerData(*section*, *orientation*, *role*)

index(*row*, *column*, *parent*)

parent(*index*)

rowCount(*parent*)

setComparison(*qcobj*)
Set comparison qcobj for highlighting differences

setData(*index*, *value*, *role*)

setupModelData(*qcobj*)
Populate model with data from QCconfigObj instance

class qcobj.cfggui.**TreeView**(**args*)
Bases: PyQt5.QtWidgets.QTreeView

__init__(**args*)

resizeColumns()

class qcobj.cfggui.**QuantityDialog**(*text*, *parent=None*)
Bases: PyQt5.QtWidgets.QDialog

__init__(*text*, *parent=None*)

class qcobj.cfggui.**CfgGui**(*opts*)
Bases: PyQt5.QtWidgets.QWidget

__init__(*opts*)

_loadQCobjs(*pn*)
Load all QConfigObj instances form file(s) in pn Remove blanks in polygons and create the widgets for every instance.

setFileChanged(*filename*)

closeEvent(*event*)

openFile()

saveFile()

toggleExpand(**args*)

6.3.3 qtCompat Module

The main initialization of our PySide/PyQt4/pyQt5 Package.

Warning: This module tries to import PySide if available, otherwise defaults to PyQt4/PyQt5 for the GUI. To change this behaviour set *_TRY_PYSIDE* to False.

Author

- 2009-2011 Nicola Creati
- 2009-2018 Roberto Vidmar

Copyright 2011-2012 Nicola Creati <ncreati@inogs.it> Roberto Vidmar <rvidmar@inogs.it>

License MIT/X11 License (see `license.txt`)

qcobj.qtCompat.**_pyside_import_module**(*moduleName*)
The import for PySide

qcobj.qtCompat.**_pyqt4_import_module**(*moduleName*)
The import for PyQt4

qcobj.qtCompat.**import_module**(*moduleName*)
The import for PyQt4

qcobj.qtCompat.**getOpenFileName**(**args*, ***kargs*)
Wrap to PyQt4 QtWidgets.QFileDialog.getOpenFileName

qcobj.qtCompat.**getOpenFileNames**(**args*, ***kargs*)
Wrap to PyQt4 QtWidgets.QFileDialog.getOpenFileNames

qcobj.qtCompat.**getSaveFileName**(**args*, ***kargs*)
Wrap to PyQt4 QtWidgets.QFileDialog.getSaveFileName

6.4 fast19.cfg

```
# Created by cfggui.py 20170412 at 14:56:38

# Description of this simulation
description = fast19

# Number of Tracers along X (width)
NumberOfTracersAlongX = 2000
# Number of Tracers along Y (depth)
NumberOfTracersAlongY = 800
# Viscosity limits for rocks, Pa
etamin = 1e17 pascal * second     # Lower limit, Pa
# Viscosity maximum limit for rocks
etamax = 1.0e25 pascal * second     # Upper limit, Pa

# Lower stress limit for power law, Pa
```

```
stressmin = 10.0e3 pascal

# Viscoelastic timestep, years
timemax = 1.0e4 year
# timemax = 30.0e3 * year

# Maximal tracers displacement step, number of gridsteps
tracerMaxMove = 0.3

# Moving Tracers:
tracerMoveAlg = runge-kutta

# Maximal temperature change, allowed for one timestep, K
tempmax = 20.0 kelvin

# Number of timesteps
stepmax = 4001

# Ouput file
output_file = fast19.hdf5
# Log file
log_file = fast19.log

num_threads = 1

# Visco-elasto-plastic iteration
vep_iteration = local
# Maximum number of Visco-elasto-plastic iterations
vep_max_iter = 10

# Tracers interpolation to image
tracers2image = False

# Temperature solver options
# temperatureSolver = '''
#     ksp_type=preonly
#     pc_type=lu
#     pc_factor_mat_solver_package=superlu_dist
#     mat_superlu_dist_colperm=PARMETIS
#     mat_superlu_dist_parsymbfact=1
#     '''
temperatureSolver = '''
    ksp_type=preonly
    pc_type=lu
    pc_factor_mat_solver_package=mumps
    mat_mumps_icntl_14=70
    '''
#temperatureSolver = '''
    #ksp_type=preonly
    #pc_type=lu
    #pc_factor_mat_solver_package=mkl_pardiso
    #'''

# Stokes solver options
# stokesSolver = '''
#     ksp_type=preonly
#     pc_type=lu
#     pc_factor_mat_solver_package=superlu_dist
#     mat_superlu_dist_colperm=PARMETIS
#     mat_superlu_dist_parsymbfact=1
#     '''
stokesSolver = '''
    ksp_type=preonly
```

```
    pc_type=lu
    pc_factor_mat_solver_package=mumps
    mat_mumps_icntl_14=70
    '''
#stokesSolver = '''
    #ksp_type=preonly
    #pc_type=lu
    #pc_factor_mat_solver_package=mkl_pardiso
    #'''
# Gas Constant
RGAS = 8.314 joule / kelvin / mole
# Universal gravity constant
UG = 66.72e-12 meter ** 2 * newton / kilogram ** 2
# Acceleration in x
gx = 0.0 meter / second ** 2
# Acceleration in y
gy = 9.81 meter / second ** 2
# Save every `save_every` evolution steps
save_every = 50
# Starting step
# step0 = 1331
# Use numerical Stress subgrid diffusion
stress_subgrid = True
# Numerical Stress subgrid value
stress_subgrid_value = 1.0 dimensionless
# Use numerical Temperature subgrid diffusion
temperature_subgrid = True
# Numerical Temperature subgrid value
temperature_subgrid_value = 1.0 dimensionless
# Apply shear heating
shear_heating = True
# Apply adiabatic heating
adiabatic_heating = True
# Pressure boundary conditions:
# onecell: pressure in one cell definition
# topbottom: pressure at the top and in the bottom of the channel
pressure_bc = onecell
# Cell boundary condition pressure
pressure_value = 100.0e3 pascal
# Maximum number of iterations for temperature
temp_iter_max = 100
# External file with Lagriangian Tracers
tracersFile = ""
# Starting step
# step0 = 501
# External configuration files blank separated
configFiles = ""

# Phase change switch
phase_change = False

[Melting]
    # Enable melting
    enabled = False
    model = gerya
#     # Polyon defining melting region
#     polygon = ""
#     # Melting fraction
#     fraction = 0.25
#     # Base rock lithology
#     lithology = NoLithology
[Mesh]
    [[X]]
```

```
        width = 3000.0 kilometer
        nodes = 201
        [[[Distribution]]]
            [[[[Variable]]]]
                end = 800.0 kilometer
                step = -8.0 kilometer
                nodes = 25
                # Starting point abscissa
                start = -1.0 meter
            [[[[Constant]]]]
                end = 2000.0 kilometer
                nodes = 150
                # Starting point abscissa
                start = -1.0 meter
                # Step width/depth (0 for constant distribution, <0 for right to␣
↪left)
                step = 0.0 meter
            [[[[Variable1]]]]
                step = 8.0 kilometer
                nodes = 25
                # Starting point abscissa
                start = -1.0 meter
                # Ending point abscissa
                end = -1.0 meter
    [[Y]]
        depth = 400.0 kilometer
        nodes = 81
        [[[Distribution]]]
            [[[[Constant]]]]
                end = 200.0 kilometer
                nodes = 50
                # Starting point abscissa
                start = -1.0 meter
                # Step width/depth (0 for constant distribution, <0 for right to␣
↪left)
                step = 0.0 meter
            [[[[Variable]]]]
                step = 4.0 kilometer
                nodes = 30
                # Starting point abscissa
                start = -1.0 meter
                # Ending point abscissa
                end = -1.0 meter
    [[Reference Point]]
        # Abscissa
        x = 0.0 meter
        # Depth
        y = 0.0 meter
        # Lithologies at the reference point
        lithologies = lithoalias1, lithoalias2
[Lithologies]
    [[air]]
        # AD
        AD = 0.0 1 / pascal / second
        # a
        a = 0.0 watt / meter
        # Layer alias
        alias = air
        # Cohesion 0
        cohesion_0 = 0.0 pascal
        # Cohesion 1
        cohesion_1 = 0.0 pascal
        # Compressibility
```

```
        compressibility = 10.0e-12 1 / pascal
        # Standard density
        density = 1000.0 kilogram / meter ** 3
        # Ea
        Ea = 0.0 joule / mole
        # Epsilon 0
        epsilon_0 = 0.0 dimensionless
        # Epsilon 1
        epsilon_1 = 1.0 dimensionless
        # Heat Capacity
        heat_capacity = 3000.0 joule / kilogram
        # Heterogeneity params: mean, std, min, max
        heterogeneity = 0, 0, 0, 0
        # Radiogenic heat production
        hr = 0.0 watt / meter ** 3
        # k0
        k0 = 300.0 watt / kelvin / meter
        # Melt density
        melt_density = 0.0 kilogram / meter ** 3
        # Melt Viscosity
        melt_viscosity = 0.0 pascal * second
        # n
        n = 0.0 dimensionless
        # Apply Peierls
        peierls = False
        # Phi 0
        phi_0 = 0.0 degree
        # Phi 1
        phi_1 = 0.0 degree
        # Use power law
        power_law = False
        # Rock type
        rock_type = none
        # Scaling Factor
        scaling_factor = 0.0 dimensionless
        # Scaling Factor Equation
        scaling_factor_eq = none
        # Shear modulus
        shear_modulus = 100.0e18 pascal
        # This equation controls melting
        SolidusEquation = 0 dimensionless
        # Thermal expansion
        thermal_expansion = 30.0e-6 1 / kelvin
        # Multiplyer factor for units in polygons
        units_multiplyer = 1.0 kilometer
        # Va
        Va = 0.0 meter ** 3
        # Viscosity
        viscosity = 1.0e18 pascal * second
        # Polygons defined by their vertices, one per line,
        # separated by nl
        polygons = '''
                0.0     0.0
                0.0     7.0
             1450.0     7.0
             1500.0     7.0
             3000.0     7.0
             3000.0     0.0'''
        # Phase change file
        phase_change = ""
        # Alias rock of wet status
        wet = ""
        # Alias rock of dry status
```

```
        dry = ""
        # Pore fluid pressure factor
        lambda = 1.0 dimensionless
        # Fluid model for the layer
        fluid = none
    [[sediments]]
        # Phi 0
        phi_0 = 1.7 degree
        # Phi 1
        phi_1 = 1.7 degree
        # Epsilon 1
        epsilon_1 = 1.0 dimensionless
        # Standard density
        density = 2600.0 kilogram / meter ** 3
        # AD
        AD = 320.0e-6 1 / pascal / second
        # a
        a = 807.0 watt / meter
        # Layer alias
        alias = sediments
        # Cohesion 0
        cohesion_0 = 1.0e6 pascal
        # Cohesion 1
        cohesion_1 = 1.0e6 pascal
        # Compressibility
        compressibility = 10.0e-12 1 / pascal
        # Ea
        Ea = 154.0e3 joule / mole
        # Epsilon 0
        epsilon_0 = 0.0 dimensionless
        # Heat Capacity
        heat_capacity = 1000.0 joule / kilogram
        # Heterogeneity params: mean, std, min, max
        heterogeneity = 0, 0, 0, 0
        # Radiogenic heat production
        hr = 2.0e-6 watt / meter ** 3
        # k0
        k0 = 640.0e-3 watt / kelvin / meter
        # Melt density
        melt_density = 2400.0 kilogram / meter ** 3
        # Melt Viscosity
        melt_viscosity = 0.0 pascal * second
        # n
        n = 2.3 dimensionless
        phase_change = sed300.dat
        fluid = wet
        # Apply Peierls
        peierls = False
        # Use power law
        power_law = True
        # Rock type
        rock_type = none
        # Scaling Factor
        scaling_factor = 0.0 dimensionless
        # Scaling Factor Equation
        scaling_factor_eq = none
        # Shear modulus
        shear_modulus = 10.0e9 pascal
        # This equation controls melting
        # Thermal expansion
        thermal_expansion = 30.0e-6 1 / kelvin
        # Multiplyer factor for units in polygons
        units_multiplyer = 1.0 kilometer
```

```
        # Va
        Va = 0.0 meter ** 3
        # Polygons defined by their vertices, one per line,
        # separated by nl
        polygons = '''
             1450.0     7.0
             3000.0     7.0
             3000.0    11.0
             1540.0    11.0
             1530.0    13.0
             1520.0    18.0
             1500.0    18.0
             1500.0    10.0'''
        SolidusEquation = 1 dimensionless
        # Viscosity
        viscosity = 0.0 pascal * second
        # Alias rock of wet status
        wet = ""
        # Alias rock of dry status
        dry = ""
        # Pore fluid pressure factor
        lambda = 1.0 dimensionless
    [[Oceanic Upper crust]]
        # AD
        AD = 320.0e-6 1 / pascal / second
        # a
        a = 474.0 watt / meter
        # Layer alias
        alias = oceanic upper crust
        # Cohesion 0
        cohesion_0 = 1.0e6 pascal
        # Cohesion 1
        cohesion_1 = 1.0e6 pascal
        # Compressibility
        compressibility = 10.0e-12 1 / pascal
        # Standard density
        density = 3000.0 kilogram / meter ** 3
        # Ea
        Ea = 154.0e3 joule / mole
        # Epsilon 0
        epsilon_0 = 0.0 dimensionless
        # Epsilon 1
        epsilon_1 = 1.0 dimensionless
        # Heat Capacity
        heat_capacity = 1000.0 joule / kilogram
        # Heterogeneity params: mean, std, min, max
        heterogeneity = 0, 0, 0, 0
        # Radiogenic heat production
        hr = 250.0e-9 watt / meter ** 3
        # k0
        k0 = 1.18 watt / kelvin / meter
        # Melt density
        melt_density = 2400.0 kilogram / meter ** 3
        # Melt Viscosity
        melt_viscosity = 0.0 pascal * second
        # n
        n = 2.3 dimensionless
        # Apply Peierls
        peierls = False
        phase_change = morb300.dat
        fluid = wet
        # Phi 0
        phi_0 = 1.7 degree
```

```
        # Phi 1
        phi_1 = 1.7 degree
        # Use power law
        power_law = True
        # Rock type
        rock_type = none
        # Scaling Factor
        scaling_factor = 0.0 dimensionless
        # Scaling Factor Equation
        scaling_factor_eq = none
        # Shear modulus
        shear_modulus = 25.0e9 pascal
        # This equation controls melting
        SolidusEquation = 2 dimensionless
        # Thermal expansion
        thermal_expansion = 30.0e-6 1 / kelvin
        # Multiplyer factor for units in polygons
        units_multiplyer = 1.0 kilometer
        # Va
        Va = 0.0 meter ** 3
        # Viscosity
        viscosity = 0.0 pascal * second
        # Polygons defined by their vertices, one per line,
        # separated by nl
        polygons = '''
             1530.0    13.0
             1540.0    11.0
             3000.0    11.0
             3000.0    13.0'''
        # Alias rock of wet status
        wet = ""
        # Alias rock of dry status
        dry = ""
        # Pore fluid pressure factor
        lambda = 1.0 dimensionless
    [[Oceanic Lower crust]]
        # AD
        AD = 330.0e-6 1 / pascal / second
        # a
        a = 474.0 watt / meter
        # Layer alias
        alias = oceanic lower crust
        # Cohesion 0
        cohesion_0 = 1.0e6 pascal
        # Cohesion 1
        cohesion_1 = 1.0e6 pascal
        # Compressibility
        compressibility = 10.0e-12 1 / pascal
        # Standard density
        density = 3200.0 kilogram / meter ** 3
        # Ea
        Ea = 238.0e3 joule / mole
        # Epsilon 0
        epsilon_0 = 0.0 dimensionless
        # Epsilon 1
        epsilon_1 = 1.0 dimensionless
        # Heat Capacity
        heat_capacity = 1000.0 joule / kilogram
        # Heterogeneity params: mean, std, min, max
        heterogeneity = 0, 0, 0, 0
        # Radiogenic heat production
        hr = 250.0e-9 watt / meter ** 3
        # k0
```

```
        k0 = 1.18 watt / kelvin / meter
        # Melt density
        melt_density = 2700.0 kilogram / meter ** 3
        # Melt Viscosity
        melt_viscosity = 0.0 pascal * second
        # n
        n = 3.2 dimensionless
        # Apply Peierls
        peierls = False
        fluid = wet
        # Phi 0
        phi_0 = 11.5 degree
        # Phi 1
        phi_1 = 11.5 degree
        # Use power law
        power_law = True
        # Rock type
        rock_type = none
        # Scaling Factor
        scaling_factor = 0.0 dimensionless
        # Scaling Factor Equation
        scaling_factor_eq = none
        # Shear modulus
        shear_modulus = 25.0e9 pascal
        # This equation controls melting
        SolidusEquation = 2 dimensionless
        # Thermal expansion
        thermal_expansion = 30.0e-6 1 / kelvin
        # Multiplyer factor for units in polygons
        units_multiplyer = 1.0 kilometer
        # Va
        Va = 0.0 meter ** 3
        # Viscosity
        viscosity = 0.0 pascal * second
        # Polygons defined by their vertices, one per line,
        # separated by nl
        polygons = '''
             1520.0    18.0
             1530.0    13.0
             3000.0    13.0
             3000.0    18.0'''
        # Phase change file
        phase_change = gabbro300.dat
        # Alias rock of wet status
        wet = ""
        # Alias rock of dry status
        dry = ""
        # Pore fluid pressure factor
        lambda = 1.0 dimensionless
    [[Lid]]
        # AD
        AD = 25.0e3 1 / pascal / second
        # a
        a = 1293.0 watt / meter
        # Layer alias
        alias = lid
        # Cohesion 0
        cohesion_0 = 1.0e6 pascal
        # Cohesion 1
        cohesion_1 = 1.0e6 pascal
        # Compressibility
        compressibility = 10.0e-12 1 / pascal
        # Standard density
```

```
        density = 3300.0 kilogram / meter ** 3
        # Ea
        Ea = 532.0e3 joule / mole
        # Epsilon 0
        epsilon_0 = 0.0 dimensionless
        # Epsilon 1
        epsilon_1 = 1.0 dimensionless
        # Heat Capacity
        heat_capacity = 1000.0 joule / kilogram
        # Heterogeneity params: mean, std, min, max
        heterogeneity = 0, 0, 0, 0
        # Radiogenic heat production
        hr = 22.0e-9 watt / meter ** 3
        # k0
        k0 = 730.0e-3 watt / kelvin / meter
        # Melt density
        melt_density = 2700.0 kilogram / meter ** 3
        # Melt Viscosity
        melt_viscosity = 0.0 pascal * second
        # n
        n = 3.5 dimensionless
        # Apply Peierls
        peierls = False
        # Phi 0
        phi_0 = 37.0 degree
        # Phi 1
        phi_1 = 37.0 degree
        # Use power law
        power_law = True
        # Rock type
        rock_type = none
        # Scaling Factor
        scaling_factor = 0.0 dimensionless
        # Scaling Factor Equation
        scaling_factor_eq = none
        # Shear modulus
        shear_modulus = 67.0e9 pascal
        # This equation controls melting
        SolidusEquation = 3 dimensionless
        # Thermal expansion
        thermal_expansion = 30.0e-6 1 / kelvin
        # Multiplyer factor for units in polygons
        units_multiplyer = 1.0 kilometer
        # Va
        Va = 10.0 centimeter ** 3
        # Viscosity
        viscosity = 0.0 pascal * second
        # Polygons defined by their vertices, one per line,
        # separated by nl
        polygons = '''
                0.0    42.0
             1462.0    42.0
             1350.0   137.0
                0.0   137.0

             1550.0    18.0
             3000.0    18.0
             3000.0   111.0
             1520.0   111.0
             1450.0   137.0
             1365.0   137.0
             1520.0    18.0'''
        wet = hmantle
```

```
        fluid = dry/wet
        # Phase change file
        phase_change = peridotite300.dat
        # Alias rock of dry status
        dry = ""
        # Pore fluid pressure factor
        lambda = 1.0 dimensionless
    [[Asthenosphere]]
        # AD
        AD = 25.0e3 1 / pascal / second
        # a
        a = 1293.0 watt / meter
        # Layer alias
        alias = asthenosphere
        # Cohesion 0
        cohesion_0 = 1.0e6 pascal
        # Cohesion 1
        cohesion_1 = 1.0e6 pascal
        # Compressibility
        compressibility = 10.0e-12 1 / pascal
        # Standard density
        density = 3300.0 kilogram / meter ** 3
        # Ea
        Ea = 532.0e3 joule / mole
        # Epsilon 0
        epsilon_0 = 0.0 dimensionless
        # Epsilon 1
        epsilon_1 = 1.0 dimensionless
        # Heat Capacity
        heat_capacity = 1000.0 joule / kilogram
        # Heterogeneity params: mean, std, min, max
        heterogeneity = 0, 0, 0, 0
        # Radiogenic heat production
        hr = 22.0e-9 watt / meter ** 3
        # k0
        k0 = 730.0e-3 watt / kelvin / meter
        # Melt density
        melt_density = 2700.0 kilogram / meter ** 3
        # Melt Viscosity
        melt_viscosity = 0.0 pascal * second
        # n
        n = 3.5 dimensionless
        # Apply Peierls
        peierls = False
        fluid = dry/wet
        # Phi 0
        phi_0 = 37.0 degree
        # Phi 1
        phi_1 = 37.0 degree
        # Use power law
        power_law = True
        # Rock type
        rock_type = none
        # Scaling Factor
        scaling_factor = 0.0 dimensionless
        # Scaling Factor Equation
        scaling_factor_eq = none
        # Shear modulus
        shear_modulus = 67.0e9 pascal
        # This equation controls melting
        SolidusEquation = 3 dimensionless
        # Thermal expansion
        thermal_expansion = 30.0e-6 1 / kelvin
```

```
        # Multiplyer factor for units in polygons
        units_multiplyer = 1.0 kilometer
        # Va
        Va = 10.0 centimeter ** 3
        # Viscosity
        viscosity = 0.0 pascal * second
        # Polygons defined by their vertices, one per line,
        # separated by nl
        polygons = '''
                0.0   137.0
             1350.0   137.0
             1450.0   137.0
             1520.0   111.0
             3000.0   111.0
             3000.0   400.0
                0.0   400.0'''
        # Phase change file
        phase_change = peridotite300.dat
        # Alias rock of wet status
        wet = hmantle
        # Alias rock of dry status
        dry = ""
        # Pore fluid pressure factor
        lambda = 1.0 dimensionless
    [[Weak Zone]]
        # AD
        AD = 2000.0 1 / pascal / second
        # a
        a = 1293.0 watt / meter
        # Layer alias
        alias = weak
        # Cohesion 0
        cohesion_0 = 1.0e6 pascal
        # Cohesion 1
        cohesion_1 = 1.0e6 pascal
        # Compressibility
        compressibility = 10.0e-12 1 / pascal
        # Standard density
        density = 3200.0 kilogram / meter ** 3
        # Ea
        Ea = 471.0e3 joule / mole
        # Epsilon 0
        # Epsilon 0
        epsilon_0 = 0.0 dimensionless
        # Epsilon 1
        epsilon_1 = 1.0 dimensionless
        # Heat Capacity
        heat_capacity = 1000.0 joule / kilogram
        # Heterogeneity params: mean, std, min, max
        heterogeneity = 0, 0, 0, 0
        # Radiogenic heat production
        hr = 22.0e-9 watt / meter ** 3
        # k0
        k0 = 730.0e-3 watt / kelvin / meter
        # Melt density
        melt_density = 2700.0 kilogram / meter ** 3
        # Melt Viscosity
        melt_viscosity = 0.0 pascal * second
        # n
        n = 4.0 dimensionless
        # Apply Peierls
        peierls = False
        # Phi 0
```

```
        phi_0 = 0.0 degree
        # Phi 1
        phi_1 = 0.0 degree
        # Use power law
        power_law = True
        # Rock type
        rock_type = none
        # Scaling Factor
        scaling_factor = 0.0 dimensionless
        # Scaling Factor Equation
        scaling_factor_eq = none
        # Shear modulus
        shear_modulus = 67.0e9 pascal
        # This equation controls melting
        SolidusEquation = 0 dimensionless
        # Thermal expansion
        thermal_expansion = 30.0e-6 1 / kelvin
        # Multiplyer factor for units in polygons
        units_multiplyer = 1.0 kilometer
        # Va
        Va = 0.0 meter ** 3
        # Viscosity
        viscosity = 0.0 pascal * second
        # Polygons defined by their vertices, one per line,
        # separated by nl
        polygons = '''
             1520.0    18.0
             1500.0    18.0
             1462.0    42.0
             1350.0   137.0
             1365.0   137.0'''
        # Phase change file
        phase_change = ""
        # Alias rock of wet status
        wet = ""
        # Alias rock of dry status
        dry = ""
        # Pore fluid pressure factor
        lambda = 1.0 dimensionless
        # Fluid model for the layer
        fluid = none
    [[Continental Upper crust]]
        # AD
        AD = 320.0e-6 1 / pascal / second
        # a
        a = 807.0 watt / meter
        # Layer alias
        alias = continental upper crust
        # Cohesion 0
        cohesion_0 = 1.0e6 pascal
        # Cohesion 1
        cohesion_1 = 1.0e6 pascal
        # Compressibility
        compressibility = 10.0e-12 1 / pascal
        # Standard density
        density = 2700.0 kilogram / meter ** 3
        # Ea
        Ea = 154.0e3 joule / mole
        # Epsilon 0
        epsilon_0 = 0.0 dimensionless
        # Epsilon 1
        epsilon_1 = 1.0 dimensionless
        # Heat Capacity
```

```
        heat_capacity = 1000.0 joule / kilogram
        # Heterogeneity params: mean, std, min, max
        heterogeneity = 0, 0, 0, 0
        # Radiogenic heat production
        hr = 1.0e-6 watt / meter ** 3
        # k0
        k0 = 640.0e-3 watt / kelvin / meter
        # Melt density
        melt_density = 2400.0 kilogram / meter ** 3
        # Melt Viscosity
        melt_viscosity = 0.0 pascal * second
        # n
        n = 2.3 dimensionless
        # Apply Peierls
        peierls = False
        # Phi 0
        phi_0 = 11.5 degree
        # Phi 1
        phi_1 = 11.5 degree
        # Use power law
        power_law = True
        # Rock type
        rock_type = none
        # Scaling Factor
        scaling_factor = 0.0 dimensionless
        # Scaling Factor Equation
        scaling_factor_eq = none
        # Shear modulus
        shear_modulus = 10.0e9 pascal
        # This equation controls melting
        SolidusEquation = 1 dimensionless
        # Thermal expansion
        thermal_expansion = 30.0e-6 1 / kelvin
        # Multiplyer factor for units in polygons
        units_multiplyer = 1.0 kilometer
        # Va
        Va = 0.0 meter ** 3
        # Viscosity
        viscosity = 0.0 pascal * second
        # Polygons defined by their vertices, one per line,
        # separated by nl
        polygons = '''
                0.0     7.0
                0.0    27.0
             1450.0    27.0
             1500.0    13.0
             1500.0    10.0
             1450.0     7.0'''
        # Phase change file
        phase_change = ""
        # Alias rock of wet status
        wet = ""
        # Alias rock of dry status
        dry = ""
        # Pore fluid pressure factor
        lambda = 1.0 dimensionless
        # Fluid model for the layer
        fluid = none
    [[Continental Lower crust]]
        # AD
        AD = 330.0e-6 1 / pascal / second
        # a
        a = 474.0 watt / meter
```

```
        # Layer alias
        alias = continental lower crust
        # Cohesion 0
        cohesion_0 = 1.0e6 pascal
        # Cohesion 1
        cohesion_1 = 1.0e6 pascal
        # Compressibility
        compressibility = 10.0e-12 1 / pascal
        # Standard density
        density = 2900.0 kilogram / meter ** 3
        # Ea
        Ea = 238.0e3 joule / mole
        # Epsilon 0
        epsilon_0 = 0.0 dimensionless
        # Epsilon 1
        epsilon_1 = 1.0 dimensionless
        # Heat Capacity
        heat_capacity = 1000.0 joule / kilogram
        # Heterogeneity params: mean, std, min, max
        heterogeneity = 0, 0, 0, 0
        # Radiogenic heat production
        hr = 500.0e-9 watt / meter ** 3
        # k0
        k0 = 1.18 watt / kelvin / meter
        # Melt density
        melt_density = 2700.0 kilogram / meter ** 3
        # Melt Viscosity
        melt_viscosity = 0.0 pascal * second
        # n
        n = 3.2 dimensionless
        # Apply Peierls
        peierls = False
        # Phi 0
        phi_0 = 11.5 degree
        # Phi 1
        phi_1 = 11.5 degree
        # Use power law
        power_law = True
        # Rock type
        rock_type = none
        # Scaling Factor
        scaling_factor = 0.0 dimensionless
        # Scaling Factor Equation
        scaling_factor_eq = none
        # Shear modulus
        shear_modulus = 25.0e9 pascal
        # This equation controls melting
        SolidusEquation = 2 dimensionless
        # Thermal expansion
        thermal_expansion = 30.0e-6 1 / kelvin
        # Multiplyer factor for units in polygons
        units_multiplyer = 1.0 kilometer
        # Va
        Va = 0.0 meter ** 3
        # Viscosity
        viscosity = 0.0 pascal * second
        # Polygons defined by their vertices, one per line,
        # separated by nl
        polygons = '''
                0.0    27.0
                0.0    42.0
             1462.0    42.0
             1500.0    18.0
```

```
            1500.0    13.0
            1450.0    27.0'''
        # Phase change file
        phase_change = ""
        # Alias rock of wet status
        wet = ""
        # Alias rock of dry status
        dry = ""
        # Pore fluid pressure factor
        lambda = 1.0 dimensionless
        # Fluid model for the layer
        fluid = none
    [[Hydrated mantle]]
        # AD
        AD = 2000.0 1 / pascal / second
        # a
        a = 1293.0 watt / meter
        # Layer alias
        alias = hmantle
        # Cohesion 0
        cohesion_0 = 1.0e6 pascal
        # Cohesion 1
        cohesion_1 = 1.0e6 pascal
        # Compressibility
        compressibility = 10.0e-12 1 / pascal
        # Standard density
        density = 3200.0 kilogram / meter ** 3
        # Ea
        Ea = 471.0e3 joule / mole
        # Epsilon 0
        # Epsilon 0
        epsilon_0 = 0.0 dimensionless
        # Epsilon 1
        epsilon_1 = 1.0 dimensionless
        # Heat Capacity
        heat_capacity = 1000.0 joule / kilogram
        # Heterogeneity params: mean, std, min, max
        heterogeneity = 0, 0, 0, 0
        # Radiogenic heat production
        hr = 22.0e-9 watt / meter ** 3
        # k0
        k0 = 730.0e-3 watt / kelvin / meter
        # Melt density
        melt_density = 2700.0 kilogram / meter ** 3
        # Melt Viscosity
        melt_viscosity = 0.0 pascal * second
        # n
        n = 4.0 dimensionless
        # Apply Peierls
        peierls = False
        # Phi 0
        phi_0 = 1.7 degree
        # Phi 1
        phi_1 = 1.7 degree
        # Use power law
        power_law = True
        # Rock type
        rock_type = none
        # Scaling Factor
        scaling_factor = 0.0 dimensionless
        # Scaling Factor Equation
        scaling_factor_eq = none
        # Shear modulus
```

```
        shear_modulus = 67.0e9 pascal
        # This equation controls melting
        SolidusEquation = 4 dimensionless
        # Thermal expansion
        thermal_expansion = 30.0e-6 1 / kelvin
        # Multiplyer factor for units in polygons
        units_multiplyer = 1.0 kilometer
        # Va
        Va = 0.0 meter ** 3
        # Viscosity
        viscosity = 0.0 pascal * second
        # Phase change file
        phase_change = peridotite300.dat
        # Alias rock of wet status
        wet = ""
        # Alias rock of dry status
        dry = ""
        # Pore fluid pressure factor
        lambda = 1.0 dimensionless
        # Fluid model for the layer
        fluid = wet
        # Polygons defined by their vertices, one per line,
        # separated by nl
        polygons = ""
[Geothermic Model]
    # Bottom temperature
    T1 = 1465.0 degC
    # Surface temperature
    T0 = 0.0 degC
    [[geotherm0]]
        # Geothermic Model type
        model = constant
        # Constant layer temperature
        value = 0.0 degC
        # Polygon defined by their vertices, one per line, separated by nl
        polygon = '''
            0. 0.
            0. 7.
            1450. 7.
            1520. 11.
            3000. 11.
            3000. 0.
            '''
        # Layer age
        age = 0.0 year
        # Thermal gradient
        dtz = 0.0 kelvin / kilometer
        # Thermal disturbance
        dt = 0.0 kelvin
        # Thermal diffusivity
        kappa = 0.0 meter ** 2 / second
        # Radius of anomaly
        radius = 0.0 kilometer
        # Direction of anomaly is right to left
        rtol = False
        # Surface temperature of layer
        T0 = 0.0 degC
        # Bottom temperature of layer
        T1 = 0.0 degC
        # List / tuple of temperature values with units enclosed in quotes
        Ti = 0.0 degC, 100.0 degC
        # Center position of thermal anomaly: x
        x0 = 0.0 kilometer
```

```
        # Center position of thermal anomaly: y
        y0 = 0.0 kilometer
        # Thermal Plate thickness
        yL = 0.0 kilometer
        # Radiogenic production
        H = 0.0 meter ** 2 * watt
        # Length scale of radiogenic production
        hr = 0.0 kilometer
        # Thermal conductivity
        kd = 0.0 watt / kelvin / meter
        # Surface heat flow
        Qs = 0.0 watt / meter ** 2
        # Min oceanic depth
        ocemin = 0.0 meter
    [[geotherm1]]
        # Geothermic Model type
        model = half-space
        # Layer age
        age = 60.0e6 year
        # Thermal Plate thickness
        yL = 111.0 kilometer
        # Thermal diffusivity
        kappa = 1.0e-6 meter ** 2 / second
        # Surface temperature of layer
        T0 = 0.0 degC
        # Polygon defined by their vertices, one per line, separated by nl
        polygon = '''
                1520. 11.
                1520. 111.
                3000. 111.
                3000. 11.
                 '''
        # Thermal gradient
        dtz = 0.0 kelvin / kilometer
        # Thermal disturbance
        dt = 0.0 kelvin
        # Radius of anomaly
        radius = 0.0 kilometer
        # Direction of anomaly is right to left
        rtol = False
        # Bottom temperature of layer
        T1 = 0.0 degC
        # List / tuple of temperature values with units enclosed in quotes
        Ti = 0.0 degC, 100.0 degC
        # Constant layer temperature
        value = 0.0 degC
        # Center position of thermal anomaly: x
        x0 = 0.0 kilometer
        # Center position of thermal anomaly: y
        y0 = 0.0 kilometer
        # Radiogenic production
        H = 0.0 meter ** 2 * watt
        # Length scale of radiogenic production
        hr = 0.0 kilometer
        # Thermal conductivity
        kd = 0.0 watt / kelvin / meter
        # Surface heat flow
        Qs = 0.0 watt / meter ** 2
        # Min oceanic depth
        ocemin = 0.0 meter
    [[geotherm2]]
        model = continent
        polygon = '''
```

```
                0. 7.
                1450. 7.
                1450. 42
                0. 42.
                '''
        T0 = 0.0 degC
        yL = 137.0 kilometer
        # Layer age
        age = 0.0 year
        # Thermal gradient
        dtz = 0.0 kelvin / kilometer
        # Thermal disturbance
        dt = 0.0 kelvin
        # Thermal diffusivity
        kappa = 0.0 meter ** 2 / second
        # Radius of anomaly
        radius = 0.0 kilometer
        # Direction of anomaly is right to left
        rtol = False
        # Bottom temperature of layer
        T1 = 500.0 degC
        # List / tuple of temperature values with units enclosed in quotes
        Ti = 0.0 degC, 100.0 degC
        # Constant layer temperature
        value = 0.0 degC
        # Center position of thermal anomaly: x
        x0 = 0.0 kilometer
        # Center position of thermal anomaly: y
        y0 = 0.0 kilometer
        # Radiogenic production
        H = 0.0 meter ** 2 * watt
        # Length scale of radiogenic production
        hr = 0.0 kilometer
        # Thermal conductivity
        kd = 0.0 watt / kelvin / meter
        # Surface heat flow
        Qs = 0.0 watt / meter ** 2
        # Min oceanic depth
        ocemin = 0.0 meter
    [[geotherm3]]
        model = continent
        polygon = '''
                0. 42.
                1450. 42.
                1450. 137
                0. 137.
                '''
        T0 = 500.0 degC
        yL = 137.0 kilometer
        # Layer age
        age = 0.0 year
        # Thermal gradient
        dtz = 0.0 kelvin / kilometer
        # Thermal disturbance
        dt = 0.0 kelvin
        # Thermal diffusivity
        kappa = 0.0 meter ** 2 / second
        # Radius of anomaly
        radius = 0.0 kilometer
        # Direction of anomaly is right to left
        rtol = False
        # Bottom temperature of layer
        T1 = 0.0 degC
```

```
        # List / tuple of temperature values with units enclosed in quotes
        Ti = 0.0 degC, 100.0 degC
        # Constant layer temperature
        value = 0.0 degC
        # Center position of thermal anomaly: x
        x0 = 0.0 kilometer
        # Center position of thermal anomaly: y
        y0 = 0.0 kilometer
        # Radiogenic production
        H = 0.0 meter ** 2 * watt
        # Length scale of radiogenic production
        hr = 0.0 kilometer
        # Thermal conductivity
        kd = 0.0 watt / kelvin / meter
        # Surface heat flow
        Qs = 0.0 watt / meter ** 2
        # Min oceanic depth
        ocemin = 0.0 meter
    [[geotherm4]]
        model = interpolated
        Ti = 0.0 degC, 500.0 degC,  562.18 degC, 0.0   degC
        polygon = '''
                  1450. 7.
                  1450. 42.
                  1520. 42.
                  1520. 11.
                  '''
    [[geotherm5]]
        model = interpolated
        Ti = 500.0 degC, 1330.0 degC,  1330.0 degC, 1079.11 degC, 562.18 degC
        polygon = '''
                  1450. 42.
                  1450. 137.
                  1520. 111.
                  1520. 80.0
                  1520. 42.0
                  '''
[Boundary Conditions]
    [[Left]]
        vx = 0.0 centimeter / year
        vy = 0.0 centimeter / year
        # All velocity components 0 on the boundary
        noslip = False
        # Constant Temperature
        T = 0.0 degC
        # Simmetric Temperature (no heat flow)
        simmetricT = True
        # Constant Pressure
        P = 0.0 pascal
        # Starting evolution step for this set of parameters
        start = 0
    [[Right]]
        vx = -2.0 centimeter / year
        vy = 0.0 centimeter / year
        # All velocity components 0 on the boundary
        noslip = False
        # Constant Temperature
        T = 0.0 degC
        # Simmetric Temperature (no heat flow)
        simmetricT = True
        # Constant Pressure
        P = 0.0 pascal
        # Starting evolution step for this set of parameters
```

```
        start = 0
    [[Top]]
        vx = 0.0 centimeter / year
        vy = 0.0 centimeter / year
        # All velocity components 0 on the boundary
        noslip = False
        # Constant Temperature
        T = 0.0 degC
        # Simmetric Temperature (no heat flow)
        simmetricT = False
        # Constant Pressure
        P = 0.0 pascal
        # Starting evolution step for this set of parameters
        start = 0
    [[Bottom]]
        vx = 0.0 centimeter / year
        vy = 8.45016e-11 meter / second
        T = 1465.0 degC
        # All velocity components 0 on the boundary
        noslip = False
        # Simmetric Temperature (no heat flow)
        simmetricT = False
        # Constant Pressure
        P = 0.0 pascal
        # Starting evolution step for this set of parameters
        start = 0
[Topography]
    # Enable topography
    enabled = False
    # Water level ???
    waterlevel = 0.0 meter
    # Topography resolution (number of nodes)
    nx = 0
    # Length of topography
    size = 0.0 meter
[Peierls]
    # Sigma ???
    sigma = 9.1e9 pascal
    # A ???
    A = 63.0e-6 1 / pascal ** 2 / second
    # m ???
    m = 1.0 dimensionless
    # n ???
    n = 2.0 dimensionless
    # Critical Pressure
    Pr_crit = 200.0e6 pascal
    # Critical Temperature
    T_crit = 1100.0 degC
[Fluid]
    enabled = True
    threshold = 0.1
    velocity_model = Darcy
    rheology_model = DryWet
    # minimum depth of fluid tracers
    minDepth = 7.0e3 meter
    [[Darcy]]
        # Porosity
        porosity = 0.01 dimensionless
        # Permeability
        permeability = 1e-18 meter ** 2
        # Fluid phase density
        fluid_density = 1000.0 kilogram / meter ** 3
        # Fluid phase viscosity
```

```
        fluid_viscosity = 1.0e-4 pascal * second
```

Warning: This code has been tested *only* on Linux (Ubuntu 14.04.5 LTS and Kubuntu 17.10) but should work also on Mac and Windows (Xp and greater).

Warning: This is work in progress!

CHAPTER

SEVEN

INDICES AND TABLES

- genindex
- modindex

PYTHON MODULE INDEX

q

INDEX

Symbols

A

C

D

E

F

G

H

I

M

N

O

P

Q

R

S

T

V

W