auxi User Manual *Release 0.3.3*

Ex Mente Technologies (Pty) Ltd.

CONTENTS

1	Introduction		1
2	Getting Started 2.1 auxi Installation		3
3	Structure 3.1 Tools		
4	auxi Reference 4.1 auxi package		19 19
5	Indices and tables	11	۱7
Ру	thon Module Index	11	19
In	lex	12	21

CHAPTER

ONE

INTRODUCTION

auxi is a toolkit to help metallurgical process engineers with their day-to-day tasks. Many of the calculations that we do require things like molar masses, conversion of one compound to another using stoichiometry, heat transfer calculations, mass balances, energy balances, etc. It is usually quite time consuming to get started with these calculations in a tool like Excel. auxi aims to save you time by making many of these calculations available from within python.

We hope that auxi will help you spend less time focusing on searching for formulas and data, and setting up calculations, and more on thinking about the problems that you need to solve with these calculations. Enjoy!

For video tutorials on using auxi visit auxi youtube.

CHAPTER

TWO

GETTING STARTED

2.1 auxi Installation

auxi runs on both Linux and Windows.

2.1.1 Prerequisite

NumPy is required to run the auxi.modelling.materials packages. You can follow the instructions at http://www.scipy.org/install.html on how to install NumPy.

2.1.2 Installation

To install auxi:

```
* On Linux: sudo pip install auxi
* On Windows: pip install auxi
```

To uninstall auxi:

```
* On Linux: sudo pip uninstall auxi
* On Windows: pip uninstall auxi
```

2.2 Importing auxi Components

If you want to use auxi in one of your python modules, you need to import its components in the same way that you do for any other python package. For example, to use the stoichiometry tool, you will have to do the following:

```
from auxi.tools.chemistry import stoichiometry
```

The same method is used for all modules, functions and classes in auxi. Here are a few more import examples:

2.3 Getting Help

You can use Python's standard help function on any of auxi's components. For example:

```
import auxi
help(auxi)

from auxi.tools.chemistry import stoichiometry
help(stoichiometry)
help(stoichiometry.molar_mass)
help(stoichiometry.convert_compound)
```

All the help information that you are able to access in this way are also available through auxi's HTML documentation that is included in the auxi Python package distribution.

THREE

STRUCTURE

auxi is a Python package, and it is currently divided into two main sub-packages, namely tools and modelling. The tools package provides help with simple tasks like calculating molar masses, enthalpies of heat capacities. The modelling package helps you to do more sophisticated calculations, and to build process models.

3.1 Tools

This package currently only contains a single set of tools, namely for chemistry calculations.

3.1.1 Chemistry Tools

The auxi.tools.chemistry package contains modules, functions and classes for doing chemical calculations. Specifically, the package contains a module called "stoichiometry" for doing stoichiometry calculations, and another called "thermochemistry" for doing thermochemical calculations.

3.1.2 Chemical Calculations

The auxi.tools.chemistry sub-package provides you with modules, classes and functions to do chemical calculations. The calculations are divided into the categories of stoichiometry and thermochemistry.

Thermochemical Calculations

Preparing Thermochemical Data

The auxi.tools.chemistry.thermochemistry module provides a number of useful functions for doing thermochemical calculations that would otherwise have been quite cumbersome to do. To make these calculations possible, some thermochemical data is needed. The <code>auxi</code> distribution package currently contains data for around 80 compounds. This may, however, not be sufficient for your process calculations. FactSage data can be converted into

auxi thermochemical data using the convert_fact_file_to_auxi_thermo_file()
function.

To prepare your own compound data files with FactSage, follow these steps:

- Open FactSage.
- Click on the "View Data" button.
- Select the "Compound" option, NOT "Solution".
- Select the database that you want to use. "FactPS" should be OK.
- Type the formula of the compound you need in the box at the bottom.
- · Click OK.
- Click on the "Cp(T)" tab.
- Select "File" from the menu and then "Save As ...".
- Select the folder where you want to store all your thermochemical data files.
- The filename must have a specific format. Taking "Ar" as an example, use "Compound_Ar.txt" for the file name.
- Click "Save".

You will have to repeat this procedure for all the compounds that you need to include in your calculations.

To convert the factsage file to an auxi thermochemical file use the following code:

Loading Thermochemical Data

If you are going to use the default set of data provided with auxi, you do not need to do anything. The entire data set will be available by default. You can obtain a list of all the compounds and their phases by using the following code:

```
from auxi.tools.chemistry import thermochemistry as thermo
thermo.list_compounds()
```

Here are the first few lines of the result:

```
Compounds currently loaded in the thermochemistry module:

Ag ['G', 'L', 'S']

Ag20 ['S']

Al ['G', 'L', 'S']

Al203 ['G', 'L', 'S1', 'S2', 'S3', 'S4']
```

```
Al4C3 ['S1']
C ['G', 'S1', 'S2']
C2H2 ['G']
CH4 ['Aq', 'G']
CO ['G']
CO2 ['G']
```

The result lists all the compounds with the phases for which data are available. Taking the compound SiO2 as an example, data are available for eight solid phases (S1 to S8), for the liquid phase and for the gas phase.

If you have decided to create you own data folder, you can force auxi to use the data in that folder. Here is the code for this:

```
from auxi.tools.chemistry import thermochemistry as thermo
thermo.load_data('/home/someuser/thermodata')
thermo.list_compounds()
```

This example data folder only contains a small selection of files:

```
Compounds currently loaded in the thermo module:

Ag ['G', 'L', 'S']

CaO ['G', 'L', 'S']

Cr2O3 ['L', 'S']

Cu ['G', 'L', 'S']

CuO ['G', 'S']
```

Calculating Heat Capacity

The Cp () function in the auxi.tools.chemistry.thermochemistry module can be used to calculate the heat capacity at constant pressure for a compound. This can be done as follows:

```
from auxi.tools.chemistry import thermochemistry as thermo

Cp_H2O = thermo.Cp("H2O[L]", 70.0)
print("The Cp of 1 kg of water at 70 °C is", Cp_H2O, "kWh/K.")

Cp_H2O = thermo.Cp("H2O[G]", 70.0)
print("The Cp of 1 kg of water vapour at 70 °C is", Cp_H2O, "kWh/K.

...")

m_Zro2 = 2.34
Cp_Zro2 = thermo.Cp("Zro2[S1]", 893.5, m_Zro2)
print("The Cp of 2.34 kg of Zro2[S1] at 893.5 °C is", Cp_Zro2, "kWh/
...")
```

3.1. Tools 7

Here are the results:

```
The Cp of 1 kg of water at 70 °C is 0.0011634065724223574 kWh/K.

The Cp of 1 kg of water vapour at 70 °C is 0.0005217114220395267

→ kWh/K.

The Cp of 2.34 kg of ZrO2[S1] at 70 °C is 0.0004084615851157184 kWh/

→ K.
```

The first parameter to the function must specify both the compound's formula and phase. If the phase is not specified it is impossible to calculate a result. The heat capacity of water is clearly significantly different from that of water vapour.

The last parameter of the Cp () is mass and it is optional. If no value is specified, it is taken to be 1 kg. This was the case for the first two calculations above. A mass of 2.34 kg was specified in the last Cp calculation.

Calculating Enthalpy

The H () function in thermochemistry is used to calculate the enthalpy of a compound. This can be done as follows:

```
from auxi.tools.chemistry import thermochemistry as thermo

H_H2O = thermo.H("H2O[L]", 70.0)
print("The enthalpy of 1 kg of water at 70 °C is", H_H2O, "kWh.")

H_H2O = thermo.H("H2O[G]", 70.0)
print("The enthalpy of 1 kg of water vapour at 70 °C is", H_H2O,
→"kWh.")

m_ZrO2 = 2.34
H_ZrO2 = thermo.H("ZrO2[S1]", 893.5, m_ZrO2)
print("The enthalpy of 2.34 kg of ZrO2[S1] at 893.5 °C is", H_ZrO2,
→"kWh.")
```

Here are the results:

```
The enthalpy of 1 kg of water at 70 °C is -4.35495670039936 kWh. The enthalpy of 1 kg of water vapour at 70 °C is -3. \rightarrow 7054553712406264 kWh. The enthalpy of 2.34 kg of ZrO2[S1] at 893.5 °C is -5. \rightarrow 463105585819936 kWh.
```

The parameters to the H () function works the same as that of the Cp () function. Both formula and phase are required in the first parameter, the second is temperature in C and the third is mass, which is optional with a default value of 1 kg.

Calculating Entropy

The S() function in thermochemistry is used to calculate the entropy of a compound. This can be done as follows:

Here are the results:

```
The entropy of 1 kg of water at 70 °C is 0.0012418035680941087 kWh/

K.

The entropy of 1 kg of water vapour at 70 °C is 0.

0029829908763826032 kWh/K.

The entropy of 2.34 kg of ZrO2[S1] at 893.5 °C is 0.

000762164298048799 kWh/K.
```

The parameters to the S() function works the same as that of the Cp() function. Both formula and phase are required in the first parameter, the second is temperature in C and the third is mass, which is optional with a default value of 1 kg.

Calculating Gibbs Free Energy

The G() function in thermochemistry is used to calculate the Gibbs free energy of a compound. This can be done as follows:

3.1. Tools 9

Here are the results:

```
The Gibbs free energy of 1 kg of water at 70 °C is
-4.781081594790853 kWh.

The Gibbs free energy of 1 kg of water vapour at 70 °C is
-4.729068690471317 kWh.

The Gibbs free energy of 2.34 kg of ZrO2[S1] at 893.5 °C is
-6.352284564138569 kWh.
```

The parameters to the G () function works the same as that of the Cp () function. Both formula and phase are required in the first parameter, the second is temperature in C and the third is mass, which is optional with a default value of 1 kg.

3.1.3 Material Physical Properties Tools

The auxi.tools.transportphenomena package provides tools to calculate material physical properties.

3.1.4 Transport Phenomena Tools

The auxi.tools.transportphenomena package provides tools to do calculations related to heat transfer, mass transfer, and fluid flow.

3.1.5 Heat transfer Calculations

The auxi.tools.transportphenomena sub-package provides tools to do calculations related to heat transfer.

3.2 Modelling Frameworks

This package contains modules, functions and classes for developing different types of computational models.

3.2.1 Process Modelling

The auxi.modelling.process package contains modules, functions and classes for performing process modelling. Specifically, the package contains a module called "materials" for modelling materials.

Material Modelling

The auxi.modelling.process sub-package provides you with modules, classes and functions to model process. The process modelling package contains material models.

Material Modelling

The auxi.modelling.process.materials package contains modules, functions and classes that help perform material-related calculations and build process models. Specifically, the package contains a module called "chemistry" for a material and material package that can perform chemistry calculations, another called "psd" for a particulate material and material package that can do size distribution as well as a material and material package that can, in addition, perform slurry calculations, and another called "thermochemistry" that contains a material consisting of multiple chemical compounds, having the ability to do thermochemical calculations.

Material Models

The auxi.modelling.process.materials sub-package provides you with modules, classes and functions to do material modelling. The material models are divided into chemical, psd and thermochemistry.

Concepts

Motivation

The material, material assay and material package concepts used in auxi.modelling.process.materials may initially seem somewhat foreign to new users. These concepts were developed to assist process engineers when doing metallurgical calculations, and while developing process models. It aims to reduce the complexity and time involved in performing these important but sometimes tedious tasks. Once these concepts have been mastered, they become incredibly powerful in the hands of a metallurgical process engineer.

Materials, Material Assays and Material Packages

auxi.modelling.process.materials includes a number of different representations of materials, material assays and material packages, each of which is contained in a separate Python module. The different modules cater for different situations as follows:

• auxi.modelling.process.materials.psd describes materials using particle size distributions. It can be used for processes in which particle size is the most important material property, such as a comminution circuit.

- auxi.modelling.process.materials.slurry adds water to psd. It can describe the solid and liquid portion of a particulate process such as a comminution circuit.
- auxi.modelling.process.materials.chem can be used for doing mass balances in chemically reactive processes such as leaching, precipitation, direct reduction and smelting. Its material class describes a material using its chemical composition. This module cannot perform any energy balance calculations.
- auxi.modelling.process.materials.thermo adds thermochemistry to chem. It can be used to do mass and energy balances in chemically reactive system such as smelting furnaces, direct reduction kilns, etc.

The auxi.modelling.process.materials.thermo module will be used to illustrate the concepts here.

Material

A Material class is used to represent a "type of material". Examples are ilmenite, iron ore, coal, ferrochrome alloy, etc. These terms are fairly abstract and generic, since they don't refer to something specific. The thermo module's Material class uses uses a list of specific phases of chemical compounds to describe a "type of material". Here are some examples:

======================================		Material		
Name	Ilmenite	======== Name	Coal	
Compound		Compound		
Al203[S1]		C[S1]		
CaO[S]		H2[G]		
Cr203[S]		O2[G]		
Fe203[S1]		N2[G]		
Fe304[S1]		S[S1]		
FeO[S]		Al203[S1]		
K20[S]		CaO[S]		
Mg0[S]		Fe203[S1]		
MnO[S]		Mg0[S]		
Na20[S1]		SiO2[S1]		
P4010[S]		=========		
SiO2[S1]				
TiO2[S1]				
V205[S]				

With the Ilmenite material we are specifying that, in our model or calculation, ilmenites will consist of the 14 compounds inluded in the first list. In the case of Coal, different coals will consist of the 10 compounds listed in the second list.

Material Assay

When we need to develop a model or do some calculations, it is not sufficient to simply know that a "type of material", such as ilmenite or coal, can consist of a specified list of compound phases. We need to know what the composition of a "specific material" is. With this composition we will be able to get started on some calculations. This is where material assays come in. In the next example, assays were added to the two materials defined above:

Material			
Name	Ilmenite		
Composition Details	(mass fractions) IlmeniteA	IlmeniteB	IlmeniteC
Al203[S1]	1.16000000e-02	1.55000000e-02	9.41000000e-03
CaO[S]	2.20000000e-04	1.00000000e-05	1.70000000e-04
Cr203[S]	8.00000000e-05	2.20000000e-04	1.10000000e-04
Fe203[S1]	2.02000000e-01	4.73000000e-01	4.96740000e-01
Fe304[S1]	0.00000000e+00	0.00000000e+00	0.00000000e+00
FeO[S]	2.79000000e-01	1.91000000e-01	0.00000000e+00
K20[S]	4.00000000e-05	1.00000000e-05	5.00000000e-05
MgO[S]	1.04000000e-02	5.80000000e-03	1.09000000e-02
MnO[S]	5.4000000e-03	4.80000000e-03	5.25000000e-03
Na20[S1]	7.00000000e-05	5.00000000e-05	3.10000000e-04
P4010[S]	1.00000000e-05	3.20000000e-04	1.50000000e-04
SiO2[S1]	8.50000000e-03	4.90000000e-03	1.74400000e-02
TiO2[S1]	4.77000000e-01	2.94000000e-01	4.59490000e-01
V205[S]	3.60000000e-03	8.00000000e-03	0.00000000e+00
======================================	Coal		
Composition Details Compound	(mass fractions) ReductantA	ReductantB	
C[S1]	8.40973866e-01	1.00000000e+00	
H2[G]	1.37955186e-02	0.00000000e+00	
02[G]	4.94339606e-02	0.00000000e+00	
N2[G]	6.09802120e-03	0.00000000e+00	
S[S1]	2.04933390e-03	0.00000000e+00	
A1203[S1]	1.20884160e-03	0.00000000e+00	
CaO[S]	2.94179980e-03	0.00000000e+00	
Fe203[S1]	7.85955656e-02	0.00000000e+00	
MgO[S]	1.41179360e-03	0.00000000e+00	

```
SiO2[S1] 3.49129950e-03 0.00000000e+00
```

Our Ilmenite material now has three assays associated with it, and they are named IlmeniteA, IlmeniteB and IlmeniteC. Ilmenite therefore refers to a "type of material", and IlmeniteA, IlmeniteB and IlmeniteC refer to "specific materials".

Two assays were added to our Coal material. The first, ReductantA, refers to a coal with 84 % carbon and roughly 8.5 % ash. Reductant B is pure graphite.

Material Packages

Using auxi.modelling.process we can now create a certain quantity of a "specific material" that is identified by a material and material assay. When we do this with the thermo Material class, we also have to specify pressure and temperature. The result of creating 1000 kg of IlmeniteB at 1 atm pressure and 500 °C temperature is the following:

MaterialPackage			
Material	======================================		========
Mass	1.0000000e+03	kq	
Amount	9.81797715e+00	kmol	
Pressure	1.00000000e+00	atm	
Temperature	5.00000000e+02	°C	
Enthalpy	-1.87069549e+03	kWh	
Compound Details:			
Formula	Mass	Mass Fraction	Mole Fraction
Al203[S1]	1.55371337e+01	1.55371337e-02	1.55207829e-02
CaO[S]	1.00239573e-02	1.00239573e-05	1.82066196e-05
Cr203[S]	2.20527060e-01	2.20527060e-04	1.47782739e-04
Fe203[S1]	4.74133178e+02	4.74133178e-01	3.02416515e-01
Fe304[S1]	0.00000000e+00	0.00000000e+00	0.00000000e+00
FeO[S]	1.91457584e+02	1.91457584e-01	2.71429867e-01
K20[S]	1.00239573e-02	1.00239573e-05	1.08388880e-05
Mg0[S]	5.81389521e+00	5.81389521e-03	1.46923993e-02
MnO[S]	4.81149948e+00	4.81149948e-03	6.90848565e-03
Na20[S1]	5.01197863e-02	5.01197863e-05	8.23650657e-05
P4010[S]	3.20766632e-01	3.20766632e-04	1.15084949e-04
SiO2[S1]	4.91173906e+00	4.91173906e-03	8.32630400e-03
TiO2[S1]	2.94704343e+02	2.94704343e-01	3.75840583e-01
V205[S]	8.01916581e+00	8.01916581e-03	4.49078466e-03

In the above result some of the useful work that <code>auxi.modelling.process.materials</code> does behind the scenes is already evident. The amount in kmol and the enthalpy in kWh of the material package was calculated, as were the masses and mole fractions of the compounds.

You will notice that the mass fractions in the material package is slightly different from those in the IlmeniteB material assay. This is because the assay was automatically normalised to add up to 1.0. You can switch of normalisation if that is more appropriate.

Summary

The auxi.modelling.process.materials concepts described above can be summarised as follows:

- A material provides a list of properties that describes a "type of material".
- A material assay describes a "specific material" by providing values for the listed properties.
- A material package describes a "specific quantity of material" belonging to a certain "type of material".

You may be wondering what the use of all this is. Why go through all the effort of defining materials, material assays and material packages? The next section demonstrates the power of these concepts.

Material Package Calculations

The use of materials and material packages are demonstrated here through the use of code snippets and the results produce by that code. We will be using ilmenite in the example. Firstly, let us import the :py:class:'auxi.modelling.process.materials.thermo.Material'class, create a material object and print it out:

```
from auxi.modelling.process.materials.thermo import Material
ilmenite = Material("Ilmenite", "./materials/ilmenite.txt")
print(ilmenite)
```

The material looks as follows:

Material			
Name	Ilmenite		
Composition Details Compound	(mass fractions) IlmeniteA	IlmeniteB	IlmeniteC
A1203[S1] Ca0[S] Cr203[S] Fe203[S1] Fe304[S1] Fe0[S] K20[S]	1.16000000e-02 2.20000000e-04 8.00000000e-05 2.02000000e-01 0.00000000e+00 2.79000000e-01 4.00000000e-05	1.55000000e-02 1.00000000e-05 2.20000000e-04 4.73000000e-01 0.00000000e+00 1.91000000e-01 1.00000000e-05	9.41000000e-03 1.70000000e-04 1.10000000e-04 4.96740000e-01 0.00000000e+00 0.00000000e+00

Mg0[S]	1.0400000e-02	5.80000000e-03	1.09000000e-02
MnO[S]	5.4000000e-03	4.80000000e-03	5.25000000e-03
Na20[S1]	7.0000000e-05	5.00000000e-05	3.10000000e-04
P4010[S]	1.0000000e-05	3.20000000e-04	1.50000000e-04
SiO2[S1]	8.5000000e-03	4.90000000e-03	1.74400000e-02
TiO2[S1]	4.77000000e-01	2.94000000e-01	4.59490000e-01
V205[S]	3.60000000e-03	8.00000000e-03	0.00000000e+00
			,

Creating, Adding and Extracting

Next we can use the material object (called ilmenite) to create a material package using each of the ilmenite assays:

Different masses were created from each assay (300 kg of IlmeniteA, 500.0 kg of IlmeniteB and 250.0 kg of IlmeniteC). All three packages were assigned a pressure of 1 atm, which is of no consequence. The packages were assigned temperatures of 25, 750 and 1200 °C respectively. In three short lines of code, <code>auxi.modelling.process.materials</code> did the following for us:

- Normalise the specified assay so that the mass fractions add up to 1.0. (We can choose not to do this.)
- Calculate the mass of each compound by multiplying the component mass fraction by the total package mass.
- Calculate the mass fraction of each compound.
- Calculate the mole fraction of each compound.
- Calculate the total amount (in kmol) of components in the package.
- Calculate the total enthalpy of the package by calculating the enthalpy of each compound and adding it together.

The result is as follows:

```
MaterialPackage

Material Ilmenite
```

Mana	2 0000000-100	1	
Mass Amount	3.00000000e+02 3.52817004e+00	_	
Pressure	1.00000000e+00		
Temperature	2.50000000e+01		
Enthalpy	-6.87812118e+02		
Compound Details			
Formula 	Mass	Mass Fraction	Mole Fraction
Al203[S1]	3.48725349e+00	1.16241783e-02	9.69390473e-03
CaO[S]	6.61375661e-02	2.20458554e-04	3.34280337e-04
Cr203[S]	2.40500241e-02	8.01667468e-05	4.48486990e-05
Fe203[S1]	6.07263107e+01	2.02421036e-01	1.07784066e-01
Fe304[S1]	0.0000000e+00	0.00000000e+00	0.00000000e+00
FeO[S]	8.38744589e+01	2.79581530e-01	3.30892788e-01
K20[S]	1.20250120e-02	4.00833734e-05	3.61829148e-05
Mg0[S]	3.12650313e+00	1.04216771e-02	2.19865404e-02
MnO[S]	1.62337662e+00	5.41125541e-03	6.48625791e-03
Na20[S1]	2.10437710e-02	7.01459035e-05	9.62343053e-05
P4010[S]	3.00625301e-03	1.00208434e-05	3.00142421e-06
SiO2[S1]	2.55531506e+00	8.51771685e-03	1.20540764e-02
TiO2[S1]	1.43398268e+02	4.77994228e-01	5.08901291e-01
1102[01]		0 60750061 00	1.68652807e-03
V205[S] ====================================	1.08225108e+00 ===================================	3.60750361e-03 	1.00032007e-03
V205[S] ====================================	1.08225108e+00 ===================================	3.60/50361e-03 ====================================	1.00032007e-03
V205[S] ====================================			
V205[S] ========== =========================	======================================		1.00032007e-03
V205[S] ========= ==========================	Ilmenite 5.00000000e+02 4.90898858e+00	kg kmol	1.00032007e-03
V205[S] ====================================	Ilmenite 5.00000000e+02 4.90898858e+00 1.000000000e+00	kg kmol atm	1.00032007e-03
V205[S] ====================================	Ilmenite 5.00000000e+02 4.90898858e+00 1.00000000e+02 7.500000000e+02	kg kmol atm	1.00032007e-03
V205[S] ====================================	Ilmenite 5.00000000e+02 4.90898858e+00 1.000000000e+00	kg kmol atm	1.00032007e-03
V205[S] ====================================	Ilmenite 5.00000000e+02 4.90898858e+00 1.00000000e+02 7.500000000e+02	kg kmol atm	1.00032007e-03
V205[S] ====================================	Ilmenite 5.00000000e+02 4.90898858e+00 1.00000000e+02 7.500000000e+02	kg kmol atm	
W205[S] ===================================	Ilmenite 5.000000000e+02 4.90898858e+00 1.00000000e+02 7.50000000e+02	kg kmol atm °C kWh	Mole Fraction
V205[S] ===================================	Ilmenite 5.000000000e+02 4.90898858e+00 1.00000000e+00 7.50000000e+02 -9.05451326e+02	kg kmol atm °C kWh Mass Fraction 1.55371337e-02	Mole Fraction
V205[S] ===================================	Ilmenite 5.000000000e+02 4.90898858e+00 1.00000000e+00 7.50000000e+02 -9.05451326e+02 Mass 7.76856687e+00	kg kmol atm °C kWh Mass Fraction 1.55371337e-02	Mole Fraction 1.55207829e-02
V205[S] ===================================	Ilmenite 5.000000000e+02 4.90898858e+00 1.00000000e+02 -9.05451326e+02 -9.05451326e+02 -9.76856687e+00 5.01197863e-03	kg kmol atm °C kWh Mass Fraction 1.55371337e-02 1.00239573e-05	Mole Fraction 1.55207829e-02 1.82066196e-05
V205[S] ===================================	Ilmenite 5.000000000e+02 4.90898858e+00 1.00000000e+02 -9.05451326e+02 -9.05451326e+02	kg kmol atm °C kWh Mass Fraction 1.55371337e-02 1.00239573e-05 2.20527060e-04	Mole Fraction 1.55207829e-02 1.82066196e-05 1.47782739e-04
V205[S] ===================================	Ilmenite 5.000000000e+02 4.90898858e+00 1.00000000e+02 -9.05451326e+02 Mass 7.76856687e+00 5.01197863e-03 1.10263530e-01 2.37066589e+02	kg kmol atm °C kWh Mass Fraction 1.55371337e-02 1.00239573e-05 2.20527060e-04 4.74133178e-01	Mole Fraction 1.55207829e-02 1.82066196e-05 1.47782739e-04 3.02416515e-01
V205[S] ===================================	Ilmenite 5.000000000e+02 4.90898858e+00 1.00000000e+02 -9.05451326e+02 -9.05451326e+02	kg kmol atm °C kWh 1.55371337e-02 1.00239573e-05 2.20527060e-04 4.74133178e-01 0.00000000e+00	Mole Fraction 1.55207829e-02 1.82066196e-05 1.47782739e-04 3.02416515e-01 0.00000000e+00
V205[S] ===================================	Ilmenite 5.000000000e+02 4.90898858e+00 1.00000000e+02 -9.05451326e+02 -9.05451326e+02	kg kmol atm °C kWh Mass Fraction 1.55371337e-02 1.00239573e-05 2.20527060e-04 4.74133178e-01 0.00000000e+00 1.91457584e-01	Mole Fraction 1.55207829e-02 1.82066196e-05 1.47782739e-04 3.02416515e-01 0.00000000e+00 2.71429867e-01
V205[S] ====================================	Ilmenite 5.000000000e+02 4.90898858e+00 1.00000000e+00 7.50000000e+02 -9.05451326e+02 Mass 7.76856687e+00 5.01197863e-03 1.10263530e-01 2.37066589e+02 0.00000000e+00 9.57287918e+01 5.01197863e-03	kg kmol atm °C kWh Mass Fraction 1.55371337e-02 1.00239573e-05 2.20527060e-04 4.74133178e-01 0.0000000e+00 1.91457584e-01 1.00239573e-05	Mole Fraction 1.55207829e-02 1.82066196e-05 1.47782739e-04 3.02416515e-01 0.0000000e+00 2.71429867e-01 1.08388880e-05
V205[S] ===================================	Ilmenite 5.000000000e+02 4.90898858e+00 1.00000000e+02 -9.05451326e+02 -9.05451326e+02	kg kmol atm °C kWh 1.55371337e-02 1.00239573e-05 2.20527060e-04 4.74133178e-01 0.00000000e+00 1.91457584e-01 1.00239573e-05 5.81389521e-03 4.81149948e-03	Mole Fraction 1.55207829e-02 1.82066196e-05 1.47782739e-04 3.02416515e-01 0.0000000e+00 2.71429867e-01 1.08388880e-05 1.46923993e-02
V205[S] ===================================	Ilmenite 5.000000000e+02 4.90898858e+00 1.00000000e+02 -9.05451326e+02 -9.05451326e+02 Mass	kg kmol atm °C kWh 1.55371337e-02 1.00239573e-05 2.20527060e-04 4.74133178e-01 0.00000000e+00 1.91457584e-01 1.00239573e-05 5.81389521e-03 4.81149948e-03	Mole Fraction 1.55207829e-02 1.82066196e-05 1.47782739e-04 3.02416515e-01 0.00000000e+00 2.71429867e-01 1.08388880e-05 1.46923993e-02 6.90848565e-03

TiO2[S1]	1.47352172e+02	2.94704343e-01	3.75840583e-01
V205[S]	4.00958290e+00	8.01916581e-03	4.49078466e-03
			=========
MaterialPackage			
Material	Ilmenite		
Mass	2.50000000e+02	kg	
Amount	2.40014670e+00		
Pressure	1.00000000e+00		
Temperature	1.20000000e+03		
Enthalpy	-5.25247309e+02	kWh 	
Compound Details			
Formula	Mass	Mass Fraction	Mole Fraction
Al203[S1]	2.35245295e+00	9.40981180e-03	9.61275553e-03
CaO[S]	4.24991500e-02	1.69996600e-04	3.15758164e-04
Cr203[S]	2.74994500e-02	1.09997800e-04	7.53824179e-05
Fe203[S1]	1.24182516e+02	4.96730065e-01	3.24003606e-01
Fe304[S1]	0.00000000e+00	0.0000000e+00	0.00000000e+00
FeO[S]	0.00000000e+00	0.00000000e+00	0.0000000e+00
K20[S]	1.24997500e-02	4.99990000e-05	5.52880254e-05
Mg0[S]	2.72494550e+00	1.08997820e-02	2.81687499e-02
MnO[S]	1.31247375e+00	5.24989500e-03	7.70863128e-03
Na20[S1]	7.74984500e-02	3.09993800e-04	5.20968045e-04
P4010[S]	3.74992500e-02	1.49997000e-04	5.50346434e-05
SiO2[S1]	4.35991280e+00	1.74396512e-02	3.02328445e-02
TiO2[S1]	1.14870203e+02	4.59480810e-01	5.99250982e-01
V205[S]	0.00000000e+00	0.00000000e+00	0.00000000e+00

We can now add these three packages of ilmenite together:

```
total_package = ilma_package + ilmb_package + ilmc_package
print(total_package)
```

In one line of code we did the following:

- Calculate the total mass of each component by adding up the component masses from the three original packages.
- Calculate the mass fraction of each compound.
- Calculate the mole fraction of each compound.
- Calculate the total amount (in kmol) of compounds in the package.
- Calculate the total enthalpy of the package by adding up the enthalpies from the three original packages.
- Calculate the temperature of the new package.

This new package (total_package) looks like this:

```
_____
MaterialPackage
______
                Ilmenite
Mass
                1.05000000e+03 kg
Amount
                1.08373053e+01 kmol
Pressure
                1.00000000e+00 atm
Temperature
                6.61513374e+02 °C
               -2.11851075e+03 kWh
Enthalpy
Compound Details
               Mass
Formula
                            Mass Fraction Mole Fraction
               1.36082733e+01 1.29602603e-02 1.23153413e-02
A1203[S1]
CaO[S]
               1.13648695e-01 1.08236852e-04 1.87005885e-04
Cr203[S]
               1.61813004e-01 1.54107623e-04 9.82371950e-05
Fe203[S1]
               4.21975416e+02 4.01881349e-01 2.43833300e-01
               0.0000000e+00 0.0000000e+00 0.0000000e+00
Fe304[S1]
FeO[S]
               1.79603251e+02 1.71050715e-01 2.30674699e-01
K20[S]
                2.95367407e-02 2.81302292e-05 2.89340215e-05
MqO[S]
               8.75839623e+00 8.34132975e-03 2.00516825e-02
               5.34160012e+00 5.08723821e-03 6.94823498e-03
MnO[S]
Na20[S1]
               1.23602114e-01 1.17716299e-04 1.84018059e-04
P4010[S]
               2.00888819e-01 1.91322685e-04 6.52958859e-05
SiO2[S1]
                9.37109739e+00 8.92485465e-03 1.43915687e-02
               4.05620643e+02 3.86305374e-01 4.68638424e-01
TiO2[S1]
V205[S]
                5.09183399e+00 4.84936570e-03 2.58325918e-03
______
```

We can easily extract a part of a material package into a new one. Let us remove 30 kg from the new package and store it in a new package:

```
dust_package = total_package.extract(30.0)
print(dust_package)
print(total_package)
```

By using one line of code we subtracted 30 kg of material from the original package and created a new one containing the subtracted 30 kg. All the other properties (e.g component masses, total amount and enthalpy) of the two packages were also recalculated. The extracted 30 kg package looks like this:

Enthalpy	-6.05288787e+01	kWh	
Compound Details Formula	Mass	Mass Fraction	Mole Fraction
Al203[S1]	3.88807809e-01	1.29602603e-02	1.23153413e-02
CaO[S]	3.24710557e-03	1.08236852e-04	1.87005885e-04
Cr203[S]	4.62322868e-03	1.54107623e-04	9.82371950e-05
Fe203[S1]	1.20564405e+01	4.01881349e-01	2.43833300e-01
Fe304[S1]	0.00000000e+00	0.00000000e+00	0.00000000e+00
FeO[S]	5.13152145e+00	1.71050715e-01	2.30674699e-01
K20[S]	8.43906876e-04	2.81302292e-05	2.89340215e-05
MgO[S]	2.50239892e-01	8.34132975e-03	2.00516825e-02
MnO[S]	1.52617146e-01	5.08723821e-03	6.94823498e-03
Na20[S1]	3.53148898e-03	1.17716299e-04	1.84018059e-04
P4010[S]	5.73968055e-03	1.91322685e-04	6.52958859e-05
SiO2[S1]	2.67745640e-01	8.92485465e-03	1.43915687e-02
TiO2[S1]	1.15891612e+01	3.86305374e-01	4.68638424e-01
V205[S]	1.45480971e-01	4.84936570e-03	2.58325918e-03

The original package, which now contains 30 kg less, now looks like this:

MaterialPackage			
Material	Ilmenite		
Mass	1.02000000e+03	kg	
Amount	1.05276680e+01	kmol	
Pressure	1.00000000e+00	atm	
Temperature	6.61513374e+02	°C	
Enthalpy	-2.05798187e+03	kWh	
Compound Details			
Formula	Mass	Mass Fraction	Mole Fraction
Al203[S1]	1.32194655e+01	1.29602603e-02	1.23153413e-02
Ca0[S]	1.10401589e-01	1.08236852e-04	1.87005885e-04
Cr203[S]	1.57189775e-01	1.54107623e-04	9.82371950e-05
Fe203[S1]	4.09918976e+02	4.01881349e-01	2.43833300e-01
Fe304[S1]	0.00000000e+00	0.00000000e+00	0.00000000e+00
Fe0[S]	1.74471729e+02	1.71050715e-01	2.30674699e-01
K20[S]	2.86928338e-02	2.81302292e-05	2.89340215e-05
Mg0[S]	8.50815634e+00	8.34132975e-03	2.00516825e-02
MnO[S]	5.18898297e+00	5.08723821e-03	6.94823498e-03
Na20[S1]	1.20070625e-01	1.17716299e-04	1.84018059e-04
P4010[S]	1.95149139e-01	1.91322685e-04	6.52958859e-05
SiO2[S1]	9.10335175e+00	8.92485465e-03	1.43915687e-02
Ti02[S1]	3.94031481e+02	3.86305374e-01	4.68638424e-01
V205[S]	4.94635301e+00	4.84936570e-03	2.58325918e-03

Summary

All the other capabilities of the auxi.modelling.process.materials.thermomaterial.Materials are not demonstrated here, since the purpose of this section is simply to introduce you to the material, material assay and material package concepts in auxi.modelling.process.materials. For full details on how to use the different Material and MaterialPackage classes and objects, refer to the following section:

- section_chemistry_material_calculations
- section_psd_material_calculations
- section_psd_slurry_material_calculations
- thermochemistry material Calculations

The final point to make is that the classes in <code>auxi.modelling.process.materials</code> can assist you in perming large numbers of metallurgical calculations with very few lines of code. The purpose of this is to focus you on the process concepts rather than entagle you in the detail of tens or hundreds of stoichiometry and thermochemical calculations. This should keep your code clean and your mind clear, getting the job done well in a short space of time.

thermochemistry material Calculations

The purpose of this section is to explain a number of concepts and demonstrate the use of the Material and MaterialPackage classes in the auxi.modelling.process.materials.thermo module.

Material Description Files

You need to create one or more material description files (MDFs) before you can create a material object in Python. Material description data are stored in simple text files with ".txt" extensions. The most simple format of such a file is the "mix.txt" file shown here:

```
Compound
Al203[S1]
C[S1]
CaO[S]
Cr203[S]
Fe203[S1]
Fe304[S1]
Fe0[S]
H2[G]
K20[S]
```

```
MnO[S]
Na20[S1]
N2[G]
O2[G]
P4010[S]
S[S1]
Si02[S1]
Ti02[S1]
V205[S]
```

The file contains a header row, which in this case only contains the word "Compound". All subsequent rows contain chemical compound phases. For example, the second line contains the S1 phase of Al2O3. When you consult FactSage, you will see that S1 refers to the gamma phase. The third line contains the graphite phase of carbon, and so forth. The purpose of this file is to tell <code>auxi</code> that materials based on this MDF will contain these compound phases, and nothing else.

Material description files can also contain material assays. The content of the "ilmenite.txt" MDF is shown here:

Compound	IlmeniteA	IlmeniteB	IlmeniteC
Al203[S1]	0.01160	0.01550	0.00941
CaO[S]	0.00022	0.00001	0.00017
Cr203[S]	0.00008	0.00022	0.00011
Fe203[S1]	0.20200	0.47300	0.49674
Fe304[S1]	0.00000	0.00000	0.00000
FeO[S]	0.27900	0.19100	0.00000
K20[S]	0.00004	0.00001	0.00005
Mg0[S]	0.01040	0.00580	0.01090
MnO[S]	0.00540	0.00480	0.00525
Na20[S1]	0.00007	0.00005	0.00031
P4010[S]	0.00001	0.00032	0.00015
SiO2[S1]	0.00850	0.00490	0.01744
TiO2[S1]	0.47700	0.29400	0.45949
V205[S]	0.00360	0.00800	0.00000

The first row still contains the word "Compound" as header for the list of compound phases. The subsequent words in the first row are assay names. **An assay name may not contain space or tab characters.** If it does, it will be interpreted as more than one name.

The first column of the file has the same meaning as the single column in the "mix.txt" file. It is a list of chemical compound phases that are allowed in materials based on this MDF. All subsequent columns contain assay information. Generally the numbers are mass fractions of the different component phases for the respective material assays. If you will be normalising your assays, the numbers can be masses, percentages or mass fractions, since they will be converted to mass fractions by normalisation.

There is more twist in the MDF tale. You can add your own custom material properties to the file. The "ilmenite.txt" file was modified to include prices for the different ilmenites:

Compound	IlmeniteA	IlmeniteB	IlmeniteC
A1203[S1]	0.01160	0.01550	0.00941
CaO[S]	0.00022	0.00001	0.00017
Cr203[S]	0.00008	0.00022	0.00011
Fe203[S1]	0.20200	0.47300	0.49674
Fe304[S1]	0.00000	0.00000	0.00000
FeO[S]	0.27900	0.19100	0.00000
K20[S]	0.00004	0.00001	0.00005
Mg0[S]	0.01040	0.00580	0.01090
MnO[S]	0.00540	0.00480	0.00525
Na20[S1]	0.00007	0.00005	0.00031
P4010[S]	0.00001	0.00032	0.00015
SiO2[S1]	0.00850	0.00490	0.01744
TiO2[S1]	0.47700	0.29400	0.45949
V205[S]	0.00360	0.00800	0.00000
#			
Price[USD/ton]	47.5000	32.2300	45.1400

The name of the property in this case is "Price" and its units are "USD/ton". **There may be no spaces in the string containing the property name and units.** In this case the string is "Price[USD/ton]", which serves the purpose of describing the custom property clearly.

Be careful not to leave empty lines at the end of your material description file. It tends to cause problems.

Materials

Now that we have created a few material description files, we can create material objects in Python.

```
from auxi.modelling.process.materials.thermo import Material

ilmenite = Material("Ilmenite", "./materials/ilmenite.txt")

print(ilmenite)

reductant = Material("Reductant", "./materials/reductant.txt")

print(reductant)

mix = Material("Mix", "./materials/mix.txt")

print(mix)
```

The Material class is imported on line 1. On line 3 a Material object is created, specifying the name of the object as the first parameter, and the location and name of the material description file as the second parameter. Two more Material objects are created on lines 6 and 9. The materials are preted out after creation, with the following result:

Name	Ilmenite		
Composition Details	(mass fractions) IlmeniteA	IlmeniteB	IlmeniteC
Al203[S1]	1.16000000e-02	1.55000000e-02	9.41000000e-03
CaO[S]	2.20000000e-04	1.00000000e-05	1.70000000e-04
Cr203[S]	8.00000000e-05	2.20000000e-04	1.10000000e-04
Fe203[S1]	2.02000000e-01	4.73000000e-01	4.96740000e-01
Fe304[S1]	0.0000000e+00	0.00000000e+00	0.00000000e+00
FeO[S]	2.79000000e-01	1.91000000e-01	0.00000000e+00
K20[S]	4.0000000e-05	1.00000000e-05	5.00000000e-05
Mg0[S]	1.0400000e-02	5.80000000e-03	1.09000000e-02
MnO[S]	5.4000000e-03	4.80000000e-03	5.25000000e-03
Na20[S1]	7.0000000e-05	5.0000000e-05	3.10000000e-04
P4010[S]	1.0000000e-05	3.2000000e-04	1.50000000e-04
SiO2[S1]	8.50000000e-03	4.9000000e-03	1.74400000e-02
TiO2[S1]	4.77000000e-01	2.94000000e-01	4.59490000e-01
V205[S]	3.60000000e-03	8.00000000e-03	0.00000000e+00
Custom Properties:			
Price[USD/ton]	4.75000000e+01	3.22300000e+01	4.51400000e+01
======================================			
======================================	Reductant		
		ReductantB	
Name Composition Details Compound	(mass fractions)	ReductantB	
Name Composition Details	(mass fractions) ReductantA		
Name Composition Details Compound C[S1]	(mass fractions) ReductantA 8.40973866e-01	1.00000000e+00	
Name Composition Details Compound C[S1] H2[G] O2[G]	(mass fractions) ReductantA 8.40973866e-01 1.37955186e-02	1.00000000e+00 0.00000000e+00	
Name Composition Details Compound C[S1] H2[G] O2[G]	(mass fractions) ReductantA 8.40973866e-01 1.37955186e-02 4.94339606e-02	1.00000000e+00 0.00000000e+00 0.00000000e+00	
Name Composition Details Compound C[S1] H2[G] O2[G] N2[G] S[S1]	(mass fractions) ReductantA 8.40973866e-01 1.37955186e-02 4.94339606e-02 6.09802120e-03	1.00000000e+00 0.000000000e+00 0.000000000e+00 0.000000000e+00	
Name Composition Details Compound C[S1] H2[G] O2[G] N2[G] S[S1]	(mass fractions) ReductantA 8.40973866e-01 1.37955186e-02 4.94339606e-02 6.09802120e-03 2.04933390e-03	1.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00	
Name Composition Details Compound C[S1] H2[G] O2[G] N2[G] S[S1] A1203[S1]	(mass fractions) ReductantA 8.40973866e-01 1.37955186e-02 4.94339606e-02 6.09802120e-03 2.04933390e-03 1.20884160e-03	1.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00	
======================================	(mass fractions) ReductantA 8.40973866e-01 1.37955186e-02 4.94339606e-02 6.09802120e-03 2.04933390e-03 1.20884160e-03 2.94179980e-03	1.00000000e+00 0.000000000e+00 0.000000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00	
======================================	(mass fractions) ReductantA 8.40973866e-01 1.37955186e-02 4.94339606e-02 6.09802120e-03 2.04933390e-03 1.20884160e-03 2.94179980e-03 7.85955656e-02	1.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00	
Name	(mass fractions) ReductantA 8.40973866e-01 1.37955186e-02 4.94339606e-02 6.09802120e-03 2.04933390e-03 1.20884160e-03 2.94179980e-03 7.85955656e-02 1.41179360e-03 3.49129950e-03	1.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00	
Name	(mass fractions) ReductantA 8.40973866e-01 1.37955186e-02 4.94339606e-02 6.09802120e-03 2.04933390e-03 1.20884160e-03 2.94179980e-03 7.85955656e-02 1.41179360e-03 3.49129950e-03	1.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00	

```
Compound
A1203[S1]
C[S1]
CaO[S]
Cr203[S]
Fe203[S1]
Fe304[S1]
FeO[S]
H2[G]
K20[S]
MgO[S]
MnO[S]
Na20[S1]
N2[G]
02 [G]
P4010[S]
S[S1]
SiO2[S1]
TiO2[S1]
V205[S]
_____
```

The material objects are now ready to create material packages.

Material Packages

Creating Empty Packages

The simplest way to create material packages is to create empty ones.

```
from auxi.modelling.process.materials.thermo import Material

ilmenite = Material("Ilmenite", "./materials/ilmenite.txt")

reductant = Material("Reductant", "./materials/reductant.txt")

mix = Material("Mix", "./materials/mix.txt")

empty_ilmenite_package = ilmenite.create_package()

print(empty_ilmenite_package)

empty_reductant_package = reductant.create_package()

print(empty_reductant_package)

empty_mix_package = mix.create_package()

print(empty_mix_package)
```

The empty packages are created by calling the "create_package" method of the Material objects without passing any parameters.

	===========		=========
MaterialPackage	=======================================		==========
Material	Ilmenite		
Mass	0.00000000e+00	kq	
Amount	0.00000000e+00	-	
Pressure	1.0000000e+00	atm	
Temperature	2.50000000e+01	°C	
Enthalpy	0.00000000e+00		
Compound Details			
Formula	Mass	Mass Fraction	Mole Fraction
Al203[S1]	0.00000000e+00	0.00000000e+00	0.00000000e+00
CaO[S]	0.00000000e+00	0.00000000e+00	0.00000000e+00
Cr203[S]	0.00000000e+00	0.00000000e+00	0.00000000e+00
Fe203[S1]	0.00000000e+00	0.00000000e+00	0.00000000e+00
Fe304[S1]	0.00000000e+00	0.000000000e+00	0.00000000e+00
FeO[S]	0.00000000e+00	0.00000000e+00	0.00000000e+00
K20[S]	0.00000000e+00	0.00000000e+00	0.00000000e+00
Mg0[S]	0.00000000e+00	0.00000000e+00	0.00000000e+00
MnO[S]	0.00000000e+00	0.00000000e+00	0.00000000e+00
Ma20[S1]	0.00000000e+00	0.00000000e+00	0.00000000e+00
	0.00000000e+00	0.00000000e+00	
P4010[S]			0.00000000e+00
SiO2[S1]	0.00000000e+00	0.00000000e+00	0.00000000e+00
TiO2[S1]	0.00000000e+00	0.00000000e+00	0.00000000e+00
V205[S]	0.00000000e+00 =========	0.00000000e+00 =======	0.00000000e+00 =======
======================================			========
Material	======================================		========
Mass	0.00000000e+00	lz or	
Mass Amount	0.00000000e+00	kmol	
	1.00000000e+00		
Pressure	2.50000000e+00	atm °C	
Temperature			
Enthalpy	0.00000000e+00	KWI1	
Compound Details	Maga	Maga Properties	Molo Brooking
Formula	Mass 	Mass Fraction 	Mole Fraction
C[S1]	0.0000000e+00	0.00000000e+00	0.00000000e+00
H2[G]	0.00000000e+00	0.00000000e+00	0.00000000e+00
02 [G]	0.0000000e+00	0.0000000e+00	0.00000000e+00
N2[G]	0.00000000e+00	0.00000000e+00	0.00000000e+00
S[S1]	0.00000000e+00	0.00000000e+00	0.00000000e+00
Al203[S1]	0.00000000e+00	0.00000000e+00	0.00000000e+00
CaO[S]	0.00000000e+00	0.00000000e+00	0.00000000e+00
Fe203[S1]	0.00000000e+00	0.00000000e+00	0.00000000e+00

Mg0[S]	0.0000000e+00	0.0000000e+00	0.0000000e+00
SiO2[S1]	0.00000000e+00	0.0000000e+00	0.00000000e+00
=======================================			========
======================================			
Material	 Mix		
Mass	0.00000000e+00	kg	
Amount	0.00000000e+00	kmol	
Pressure	1.00000000e+00	atm	
Temperature	2.50000000e+01	°C	
Enthalpy	0.00000000e+00	kWh	
Compound Details			
Formula	Mass	Mass Fraction	Mole Fraction
Al203[S1]	0.00000000e+00	0.00000000e+00	0.00000000e+00
C[S1]	0.00000000e+00	0.00000000e+00	0.00000000e+00
CaO[S]	0.00000000e+00	0.0000000e+00	0.00000000e+00
Cr203[S]	0.00000000e+00	0.0000000e+00	0.00000000e+00
Fe203[S1]	0.00000000e+00	0.0000000e+00	0.00000000e+00
Fe304[S1]	0.00000000e+00	0.0000000e+00	0.0000000e+00
FeO[S]	0.00000000e+00	0.0000000e+00	0.00000000e+00
H2[G]	0.00000000e+00	0.0000000e+00	0.0000000e+00
K20[S]	0.00000000e+00	0.0000000e+00	0.00000000e+00
Mg0[S]	0.00000000e+00	0.00000000e+00	0.00000000e+00
MnO[S]	0.00000000e+00	0.0000000e+00	0.00000000e+00
Na20[S1]	0.00000000e+00	0.0000000e+00	0.00000000e+00
N2[G]	0.00000000e+00	0.0000000e+00	0.0000000e+00
02[G]	0.00000000e+00	0.0000000e+00	0.00000000e+00
P4010[S]	0.00000000e+00	0.0000000e+00	0.00000000e+00
S[S1]	0.00000000e+00	0.0000000e+00	0.00000000e+00
SiO2[S1]	0.00000000e+00	0.0000000e+00	0.00000000e+00
0102[01]		0 00000000 +00	0 00000000 .00
TiO2[S1]	0.0000000e+00	0.00000000e+00	0.00000000e+00

Creating Filled Packages

It is just as easy to create packages that contain some mass. Let's do that with ilmenite.

```
from auxi.modelling.process.materials.thermo import Material

ilmenite = Material("Ilmenite", "./materials/ilmenite.txt")

reductant = Material("Reductant", "./materials/reductant.txt")

mix = Material("Mix", "./materials/mix.txt")

ilma_package = ilmenite.create_package("IlmeniteA", 300.0, 1.0, 25.
```

```
print(ilma_package)
```

The parameters to the "create_package" method are:

- 1. material assay name, "IlmeniteA"
- 2. mass, 300 kg
- 3. pressure, 1 atm
- 4. temperature, 25 °C

We therefore created 300 kg based on the composition specified by the IlmeniteA assay, at 1 atm pressure and 25 °C temperature. The resulting package is shown here.

MaterialPackage			
Material	Ilmenite	========	=========
Mass	3.00000000e+02	kg	
Amount	3.52817004e+00	kmol	
Pressure	1.00000000e+00	atm	
Temperature	2.50000000e+01	°C	
Enthalpy	-6.87812118e+02	kWh	
Compound Details			
Formula	Mass	Mass Fraction	Mole Fraction
Al203[S1]	3.48725349e+00	1.16241783e-02	9.69390473e-03
CaO[S]	6.61375661e-02	2.20458554e-04	3.34280337e-04
Cr203[S]	2.40500241e-02	8.01667468e-05	4.48486990e-05
Fe203[S1]	6.07263107e+01	2.02421036e-01	1.07784066e-01
Fe304[S1]	0.00000000e+00	0.00000000e+00	0.00000000e+00
FeO[S]	8.38744589e+01	2.79581530e-01	3.30892788e-01
K20[S]	1.20250120e-02	4.00833734e-05	3.61829148e-05
Mg0[S]	3.12650313e+00	1.04216771e-02	2.19865404e-02
MnO[S]	1.62337662e+00	5.41125541e-03	6.48625791e-03
Na20[S1]	2.10437710e-02	7.01459035e-05	9.62343053e-05
P4010[S]	3.00625301e-03	1.00208434e-05	3.00142421e-06
SiO2[S1]	2.55531506e+00	8.51771685e-03	1.20540764e-02
TiO2[S1]	1.43398268e+02	4.77994228e-01	5.08901291e-01
V205[S]	1.08225108e+00	3.60750361e-03	1.68652807e-03

Adding Material to a Package - Another Package

Now we create another ilmenite package with a different composition, mass and temperature, and add it to the first:

```
from auxi.modelling.process.materials.thermo import Material

2
```

This changes the original "ilma_package", but the second "ilmb_package" remains the same. This is quite a powerful action, since one line of code does all of the following:

- Calculate the total mass of each component by adding up the component masses from the two packages.
- Calculate the mass fraction of each compound.
- Calculate the mole fraction of each compound.
- Calculate the total amount (in kmol) of compounds in the package.
- Calculate the total enthalpy by adding up the enthalpies of the two original packages.
- Calculate the temperature of the new package.

The resulting two packages are shown below:

=======================================	==========		========
MaterialPackage			
Material	Ilmenite		
Mass	8.00000000e+02	kg	
Amount	8.43715862e+00	kmol	
Pressure	1.00000000e+00	atm	
Temperature	4.88474167e+02	°C	
Enthalpy	-1.59326344e+03	kWh	
Compound Details			
Formula	Mass	Mass Fraction	Mole Fraction
Al203[S1]	1.12558204e+01	1.40697755e-02	1.30841549e-02
CaO[S]	7.11495448e-02	8.89369310e-05	1.50379294e-04
Cr203[S]	1.34313554e-01	1.67891942e-04	1.04738770e-04
Fe203[S1]	2.97792900e+02	3.72241125e-01	2.21026985e-01
Fe304[S1]	0.00000000e+00	0.00000000e+00	0.00000000e+00
FeO[S]	1.79603251e+02	2.24504063e-01	2.96295501e-01
K20[S]	1.70369907e-02	2.12962383e-05	2.14370100e-05
Mg0[S]	6.03345073e+00	7.54181341e-03	1.77425932e-02
MnO[S]	4.02912637e+00	5.03640796e-03	6.73192250e-03

```
Na20[S1]
                4.61036642e-02 5.76295802e-05 8.81647712e-05
P4010[S]
               1.63389569e-01 2.04236961e-04 6.82149359e-05
                5.01118458e+00 6.26398073e-03 9.88514810e-03
SiO2[S1]
TiO2[S1]
                2.90750440e+02 3.63438050e-01 4.31482633e-01
V205[S]
                5.09183399e+00 6.36479248e-03 3.31812755e-03
______
MaterialPackage
______
Material
               Ilmenite
Mass
                5.00000000e+02 kg
Amount
                4.90898858e+00 kmol
Pressure
                1.00000000e+00 atm
Temperature
                7.50000000e+02 °C
                -9.05451326e+02 kWh
Enthalpy
Compound Details
Formula
                            Mass Fraction Mole Fraction
               Mass
_____
Al203[S1]
               7.76856687e+00 1.55371337e-02 1.55207829e-02
Ca0[S]
               5.01197863e-03 1.00239573e-05 1.82066196e-05
               1.10263530e-01 2.20527060e-04 1.47782739e-04
Cr203[S]
Fe203[S1]
               2.37066589e+02 4.74133178e-01 3.02416515e-01
Fe304[S1]
               0.00000000e+00 0.0000000e+00 0.0000000e+00
FeO[S]
               9.57287918e+01 1.91457584e-01 2.71429867e-01
K20[S]
               5.01197863e-03 1.00239573e-05 1.08388880e-05
               2.90694760e+00 5.81389521e-03 1.46923993e-02
Mg0[S]
               2.40574974e+00 4.81149948e-03 6.90848565e-03
MnO[S]
Na20[S1]
               2.50598931e-02 5.01197863e-05 8.23650657e-05
               1.60383316e-01 3.20766632e-04 1.15084949e-04
P4010[S]
                2.45586953e+00 4.91173906e-03 8.32630400e-03
SiO2[S1]
TiO2[S1]
               1.47352172e+02 2.94704343e-01 3.75840583e-01
               4.00958290e+00 8.01916581e-03 4.49078466e-03
V205[S]
______
```

Adding Material to a Package - A Compound Mass

Sometimes you need to add material to a package, one compound at a time.

```
from auxi.modelling.process.materials.thermo import Material

ilmenite = Material("Ilmenite", "./materials/ilmenite.txt")

reductant = Material("Reductant", "./materials/reductant.txt")

mix = Material("Mix", "./materials/mix.txt")

ilma_package = ilmenite.create_package("IlmeniteA", 300.0, 1.0, 25.

o))
```

```
9  ilma_package += ("TiO2[S1]", 150.0)
print(ilma_package)
```

This adds 150 kg of TiO2[S1] to ilma_package. The temperature of the added material is assumed to be the same as that of the original package, which means that ilma_package's temperature does not change. Here is the result:

```
MaterialPackage
______
Material
                Ilmenite
Mass
                 4.50000000e+02 kg
Amount
                 5.40632064e+00 kmol
Pressure
                1.00000000e+00 atm
Temperature
                2.50000000e+01 °C
Enthalpy
                -1.18069622e+03 kWh
Compound Details
Formula
                             Mass Fraction Mole Fraction
                Mass
Al203[S1]
                3.48725349e+00 7.74945219e-03 6.32625154e-03
                6.61375661e-02 1.46972369e-04 2.18151669e-04
Ca0[S]
               2.40500241e-02 5.34444979e-05 2.92683040e-05
Cr203[S]
Fe203[S1]
               6.07263107e+01 1.34947357e-01 7.03399852e-02
Fe304[S1]
                0.0000000e+00 0.0000000e+00 0.0000000e+00
FeO[S]
                8.38744589e+01 1.86387686e-01 2.15940951e-01
K20[S]
                1.20250120e-02 2.67222489e-05 2.36130050e-05
                3.12650313e+00 6.94778473e-03 1.43484374e-02
Mq0[S]
                1.62337662e+00 3.60750361e-03 4.23293814e-03
MnO[S]
Na20[S1]
                2.10437710e-02 4.67639357e-05 6.28026001e-05
                3.00625301e-03 6.68056224e-06 1.95873232e-06
P4010[S]
                2.55531506e+00 5.67847790e-03 7.86650184e-03
SiO2[S1]
TiO2[S1]
                2.93398268e+02 6.51996152e-01 6.79508511e-01
V205[S]
                1.08225108e+00 2.40500241e-03 1.10062984e-03
______
```

Adding Material to a Package - A Compound Mass with Specified Temperature

We can also add a certain mass of a specified compound at a temperature different from the original package.

```
from auxi.modelling.process.materials.thermo import Material

ilmenite = Material("Ilmenite", "./materials/ilmenite.txt")

reductant = Material("Reductant", "./materials/reductant.txt")

mix = Material("Mix", "./materials/mix.txt")

ilma_package = ilmenite.create_package("IlmeniteA", 300.0, 1.0, 25.

o)
```

```
8
9 ilma_package += ("TiO2[S1]", 150.0, 1000.0)
print(ilma_package)
```

This action calculates a new total mass, component masses, mass fractions and mole fractions, as well as a new enthalpy and temperature.

MaterialPackage			
========= Material	Ilmenite		
Mass	4.50000000e+02	kg	
Amount	5.40632064e+00	kmol	
Pressure	1.00000000e+00	atm	
Temperature	3.84927151e+02	°C	
Enthalpy	-1.14449836e+03	kWh	
Compound Details			
Formula	Mass	Mass Fraction	
Al203[S1]	3.48725349e+00		
Ca0[S]	6.61375661e-02	1.46972369e-04	2.18151669e-04
Cr203[S]	2.40500241e-02	5.34444979e-05	2.92683040e-05
Fe203[S1]	6.07263107e+01	1.34947357e-01	7.03399852e-02
Fe304[S1]	0.00000000e+00	0.00000000e+00	0.00000000e+00
FeO[S]	8.38744589e+01	1.86387686e-01	2.15940951e-01
K20[S]	1.20250120e-02	2.67222489e-05	2.36130050e-05
Mg0[S]	3.12650313e+00	6.94778473e-03	1.43484374e-02
MnO[S]	1.62337662e+00	3.60750361e-03	4.23293814e-03
Na20[S1]	2.10437710e-02	4.67639357e-05	6.28026001e-05
P4010[S]	3.00625301e-03	6.68056224e-06	1.95873232e-06
SiO2[S1]	2.55531506e+00	5.67847790e-03	7.86650184e-03
TiO2[S1]	2.93398268e+02	6.51996152e-01	6.79508511e-01
V205[S]	1.08225108e+00	2.40500241e-03	1.10062984e-03

Adding Packages of Different Materials Together

We very often need to add packages from different materials together. For example, ilmenite and reductant can be added together so that reduction reactions can be modelled.

```
from auxi.modelling.process.materials.thermo import Material

ilmenite = Material("Ilmenite", "./materials/ilmenite.txt")

reductant = Material("Reductant", "./materials/reductant.txt")

mix = Material("Mix", "./materials/mix.txt")

ilma_package = ilmenite.create_package("IlmeniteA", 300.0, 1.0, 25.

o)
```

This, however, does not work. See the last two lines of the error message below.

```
Traceback (most recent call last):
File "test.py", line 10, in <module>
    new_package = ilma_package + reda_package
File "thermochemistry.material.py", line 430, in __add__
    self.material.name + "'.")
Exception: Packages of 'Reductant' cannot be added to packages of
    →'Ilmenite'.
    The compound 'C[S1]' was not found in 'Ilmenite'.
```

Let's try it by swopping the two material packages around.

```
from auxi.modelling.process.materials.thermo import Material

ilmenite = Material("Ilmenite", "./materials/ilmenite.txt")

reductant = Material("Reductant", "./materials/reductant.txt")

mix = Material("Mix", "./materials/mix.txt")

ilma_package = ilmenite.create_package("IlmeniteA", 300.0, 1.0, 25.

-0)

reda_package = reductant.create_package("ReductantA", 35.0, 1.0, 25.

-0)

new_package = reda_package + ilma_package

print(new_package)
```

```
Traceback (most recent call last):
File "test.py", line 10, in <module>
    new_package = reda_package + ilma_package
File "thermochemistry.material.py", line 430, in __add__
    self.material.name + "'.")
Exception: Packages of 'Ilmenite' cannot be added to packages of
    'Reductant'.
    The compound 'Cr203[S]' was not found in 'Reductant'.
```

Still no luck. These packages cannot be added together because their materials are not compatible. We need to use an intermediate material package from a compatible material that will allow us to add ilmenite and reductant together. This is the purpose of the "mix" material that we created early on.

```
from auxi.modelling.process.materials.thermo import Material

ilmenite = Material("Ilmenite", "./materials/ilmenite.txt")
```

```
reductant = Material("Reductant", "./materials/reductant.txt")
mix = Material("Mix", "./materials/mix.txt")

ilma_package = ilmenite.create_package("IlmeniteA", 300.0, 1.0, 25.

o)
reda_package = reductant.create_package("ReductantA", 35.0, 1.0, 25.

o)

new_package = mix.create_package()
new_package += ilma_package
new_package += reda_package
print(new_package)
```

Success at last! The mix material package is able to receive all the compound masses from both the ilmenite and reductant packages.

MaterialPackage			
Material	========== Mix	=========	=========
Mass	3.35000000e+02	kg	
Amount	6.30500835e+00	kmol	
Pressure	1.00000000e+00	atm	
Temperature	2.50000000e+01	°C	
Enthalpy	-6.92925041e+02	kWh	
Compound Details			
Formula	Mass	Mass Fraction	Mole Fraction
Al203[S1]	3.52956294e+00	1.05360088e-02	5.49034965e-03
C[S1]	2.94340853e+01	8.78629412e-02	3.88683906e-01
CaO[S]	1.69100559e-01	5.04777788e-04	4.78268203e-04
Cr203[S]	2.40500241e-02	7.17911166e-05	2.50965308e-05
Fe203[S1]	6.34771555e+01	1.89484046e-01	6.30462073e-02
Fe304[S1]	0.00000000e+00	0.00000000e+00	0.00000000e+00
FeO[S]	8.38744589e+01	2.50371519e-01	1.85161693e-01
H2[G]	4.82843151e-01	1.44132284e-03	3.79888138e-02
K20[S]	1.20250120e-02	3.58955583e-05	2.02473128e-05
Mg0[S]	3.17591590e+00	9.48034598e-03	1.24977222e-02
MnO[S]	1.62337662e+00	4.84590037e-03	3.62959406e-03
Na20[S1]	2.10437710e-02	6.28172270e-05	5.38509982e-05
N2[G]	2.13430742e-01	6.37106693e-04	1.20838199e-03
02 [G]	1.73018862e+00	5.16474215e-03	8.57578913e-03
P4010[S]	3.00625301e-03	8.97388957e-06	1.67954337e-06
S[S1]	7.17266865e-02	2.14109512e-04	3.54772799e-04
SiO2[S1]	2.67751054e+00	7.99256877e-03	7.06780432e-03
TiO2[S1]	1.43398268e+02	4.28054533e-01	2.84772072e-01
V2O5[S]	1.08225108e+00	3.23060025e-03	9.43750984e-04

Adding Material Together - Package + Package

In the above three sections we demonstrated how material can be added to an existing package. Here we will add material together to create a new package.

```
from auxi.modelling.process.materials.thermo import Material

ilmenite = Material("Ilmenite", "./materials/ilmenite.txt")

reductant = Material("Reductant", "./materials/reductant.txt")

mix = Material("Mix", "./materials/mix.txt")

ilma_package = ilmenite.create_package("IlmeniteA", 300.0, 1.0, 25.

-0)

ilmb_package = ilmenite.create_package("IlmeniteB", 500.0, 1.0, 750.

-0)

new_package = ilma_package + ilmb_package

print(new_package)
```

This action performs all the calculations to create a new package with properties based on the two original packages. Specifically note that the temperature was automatically calculated.

```
MaterialPackage
______
Material
               Ilmenite
                8.00000000e+02 kg
Mass
Amount
                8.43715862e+00 kmol
Pressure
                1.00000000e+00 atm
Temperature
                4.88474167e+02 °C
Enthalpy
                -1.59326344e+03 kWh
Compound Details
Formula
                            Mass Fraction Mole Fraction
               Mass
_____
Al203[S1]
                1.12558204e+01 1.40697755e-02 1.30841549e-02
               7.11495448e-02 8.89369310e-05 1.50379294e-04
CaO[S]
               1.34313554e-01 1.67891942e-04 1.04738770e-04
Cr203[S]
Fe203[S1]
               2.97792900e+02 3.72241125e-01 2.21026985e-01
Fe304[S1]
               0.0000000e+00 0.0000000e+00 0.0000000e+00
FeO[S]
               1.79603251e+02 2.24504063e-01 2.96295501e-01
K20[S]
               1.70369907e-02 2.12962383e-05 2.14370100e-05
                6.03345073e+00 7.54181341e-03 1.77425932e-02
Mg0[S]
MnO[S]
               4.02912637e+00 5.03640796e-03 6.73192250e-03
                4.61036642e-02 5.76295802e-05 8.81647712e-05
Na20[S1]
               1.63389569e-01 2.04236961e-04 6.82149359e-05
P4010[S]
SiO2[S1]
                5.01118458e+00 6.26398073e-03 9.88514810e-03
                2.90750440e+02 3.63438050e-01 4.31482633e-01
TiO2[S1]
V205[S]
                5.09183399e+00 6.36479248e-03 3.31812755e-03
______
```

Adding Material Together - Package + Compound Mass

Now we add a package and specific mass of a compound together to produce a new package.

```
from auxi.modelling.process.materials.thermo import Material

ilmenite = Material("Ilmenite", "./materials/ilmenite.txt")

reductant = Material("Reductant", "./materials/reductant.txt")

mix = Material("Mix", "./materials/mix.txt")

ilma_package = ilmenite.create_package("IlmeniteA", 300.0, 1.0, 25.

-0)

new_package = ilma_package + ("TiO2[S1]", 150.0)

print(new_package)
```

The added compound mass is assumed to be at the same temperature as the original package. This results in the new package having the same temperature as the original package.

MaterialPackage 			
Material	Ilmenite		
Mass	4.50000000e+02	kg	
Amount	5.40632064e+00	kmol	
Pressure	1.00000000e+00	atm	
Temperature	2.50000000e+01	°C	
Enthalpy	-1.18069622e+03	kWh	
Compound Details			
Formula	Mass	Mass Fraction	Mole Fraction
Al203[S1]	3.48725349e+00	7.74945219e-03	6.32625154e-03
Ca0[S]	6.61375661e-02	1.46972369e-04	2.18151669e-04
Cr203[S]	2.40500241e-02	5.34444979e-05	2.92683040e-05
Fe203[S1]	6.07263107e+01	1.34947357e-01	7.03399852e-02
Fe304[S1]	0.00000000e+00	0.00000000e+00	0.00000000e+00
FeO[S]	8.38744589e+01	1.86387686e-01	2.15940951e-01
K20[S]	1.20250120e-02	2.67222489e-05	2.36130050e-05
Mg0[S]	3.12650313e+00	6.94778473e-03	1.43484374e-02
MnO[S]	1.62337662e+00	3.60750361e-03	4.23293814e-03
Na20[S1]	2.10437710e-02	4.67639357e-05	6.28026001e-05
P4010[S]	3.00625301e-03	6.68056224e-06	1.95873232e-06
SiO2[S1]	2.55531506e+00	5.67847790e-03	7.86650184e-03
TiO2[S1]	2.93398268e+02	6.51996152e-01	6.79508511e-01
V205[S]	1.08225108e+00	2.40500241e-03	1.10062984e-03

Adding Material Together - Package + Compound Mass at Specified Temperature

Now we add the same compound mass as in the previous section, but at a different temperature.

```
from auxi.materials.thermo import Material

ilmenite = Material("Ilmenite", "./materials/ilmenite.txt")

reductant = Material("Reductant", "./materials/reductant.txt")

mix = Material("Mix", "./materials/mix.txt")

ilma_package = ilmenite.create_package("IlmeniteA", 300.0, 1.0, 25.

-0)

new_package = ilma_package + ("TiO2[S1]", 150.0, 1000.0)

print(new_package)
```

The new package now has a different temperature, which is calculated based on the enthalpy of the original package and the enthalpy of the added compound mass.

```
MaterialPackage
______
Material
                Ilmenite
Mass
                  4.50000000e+02 kg
Amount
                 5.40632064e+00 kmol
Pressure
                 1.00000000e+00 atm
                  3.84927151e+02 °C
Temperature
Enthalpy
                 -1.14449836e+03 kWh
Compound Details
Formula
                Mass
                               Mass Fraction Mole Fraction
______
                 3.48725349e+00 7.74945219e-03 6.32625154e-03
Al203[S1]
                 6.61375661e-02 1.46972369e-04 2.18151669e-04
Ca0[S]
                2.40500241e-02 5.34444979e-05 2.92683040e-05
Cr203[S]
Fe203[S1]
                 6.07263107e+01 1.34947357e-01
                                             7.03399852e-02
Fe304[S1]
                0.00000000e+00 0.0000000e+00 0.0000000e+00
                 8.38744589e+01 1.86387686e-01 2.15940951e-01
FeO[S]
K20[S]
                 1.20250120e-02 2.67222489e-05 2.36130050e-05
                 3.12650313e+00 6.94778473e-03 1.43484374e-02
Mg0[S]
                 1.62337662e+00 3.60750361e-03 4.23293814e-03
MnO[S]
Na20[S1]
                2.10437710e-02 4.67639357e-05 6.28026001e-05
P4010[S]
                 3.00625301e-03 6.68056224e-06 1.95873232e-06
SiO2[S1]
                 2.55531506e+00 5.67847790e-03 7.86650184e-03
                                             6.79508511e-01
TiO2[S1]
                 2.93398268e+02 6.51996152e-01
V205[S]
                 1.08225108e+00 2.40500241e-03 1.10062984e-03
```

Extract Material from a Package - Mass

When we need to create a new package by extracting material from an existing material, we use the "extract" method. First of all we can simply specify the total mass to be extracted.

```
from auxi.modelling.process.materials.thermo import Material

ilmenite = Material("Ilmenite", "./materials/ilmenite.txt")

reductant = Material("Reductant", "./materials/reductant.txt")

mix = Material("Mix", "./materials/mix.txt")

ilma_package = ilmenite.create_package("IlmeniteA", 300.0, 1.0, 25.

o))

new_package = ilma_package.extract(75.0)

print(ilma_package)

print(new_package)
```

This removes 75 kg from the original package, and produces a new package of 75 kg. The new package has the same composition, temperature and pressure as the original one.

Material	Ilmenite		
Mass	2.25000000e+02	kg	
Amount	2.64612753e+00	kmol	
Pressure	1.00000000e+00	atm	
Temperature	2.50000000e+01	°C	
Enthalpy	-5.15859089e+02	kWh	
Compound Details			
Formula	Mass	Mass Fraction	Mole Fraction
Al203[S1]	2.61544012e+00	1.16241783e-02	9.69390473e-03
CaO[S]	4.96031746e-02	2.20458554e-04	3.34280337e-04
Cr203[S]	1.80375180e-02	8.01667468e-05	4.48486990e-05
Fe203[S1]	4.55447330e+01	2.02421036e-01	1.07784066e-01
Fe304[S1]	0.00000000e+00	0.00000000e+00	0.00000000e+00
FeO[S]	6.29058442e+01	2.79581530e-01	3.30892788e-01
K20[S]	9.01875902e-03	4.00833734e-05	3.61829148e-05
Mg0[S]	2.34487734e+00	1.04216771e-02	2.19865404e-02
MnO[S]	1.21753247e+00	5.41125541e-03	6.48625791e-03
Na20[S1]	1.57828283e-02	7.01459035e-05	9.62343053e-05
P4010[S]	2.25468975e-03	1.00208434e-05	3.00142421e-06
SiO2[S1]	1.91648629e+00	8.51771685e-03	1.20540764e-02
TiO2[S1]	1.07548701e+02	4.77994228e-01	5.08901291e-01
V2O5[S]	8.11688312e-01	3.60750361e-03	1.68652807e-03

MaterialPackage			
Material	Ilmenite		
Mass	7.50000000e+01	kg	
Amount	8.82042511e-01	kmol	
Pressure	1.0000000e+00	atm	
Temperature	2.50000000e+01	°C	
Enthalpy	-1.71953030e+02	kWh	
Compound Details			
Formula	Mass	Mass Fraction	Mole Fraction
Al203[S1]	8.71813372e-01	1.16241783e-02	9.69390473e-03
CaO[S]	1.65343915e-02	2.20458554e-04	3.34280337e-04
Cr203[S]	6.01250601e-03	8.01667468e-05	4.48486990e-05
Fe203[S1]	1.51815777e+01	2.02421036e-01	1.07784066e-01
Fe304[S1]	0.0000000e+00	0.00000000e+00	0.00000000e+00
FeO[S]	2.09686147e+01	2.79581530e-01	3.30892788e-01
K20[S]	3.00625301e-03	4.00833734e-05	3.61829148e-05
Mg0[S]	7.81625782e-01	1.04216771e-02	2.19865404e-02
MnO[S]	4.05844156e-01	5.41125541e-03	6.48625791e-03
Na20[S1]	5.26094276e-03	7.01459035e-05	9.62343053e-05
P4010[S]	7.51563252e-04	1.00208434e-05	3.00142421e-06
SiO2[S1]	6.38828764e-01	8.51771685e-03	1.20540764e-02
TiO2[S1]	3.58495671e+01	4.77994228e-01	5.08901291e-01
V205[S]	2.70562771e-01	3.60750361e-03	1.68652807e-03

Extract Material from a Package - Compound

We can also extract all the mass of a single compound from an existing package into a new one.

```
from auxi.modelling.process.materials.thermo import Material

ilmenite = Material("Ilmenite", "./materials/ilmenite.txt")

reductant = Material("Reductant", "./materials/reductant.txt")

mix = Material("Mix", "./materials/mix.txt")

ilma_package = ilmenite.create_package("IlmeniteA", 300.0, 1.0, 25.

o)

new_package = ilma_package.extract("TiO2[S1]")

print(ilma_package)

print(new_package)
```

This modifies the original package's composition and enthalpy, and creates a new package of the same temperature consisting purely of the specified compound.

MaterialPackage			
========= Material	Ilmenite		=========
Mass	1.56601732e+02	kα	
Amount	1.73267975e+00	-	
Pressure	1.00000000e+00		
Temperature	2.50000000e+01		
Enthalpy	-2.16620609e+02		
 Compound Details			
Formula 	Mass	Mass Fraction	Mole Fraction
Al203[S1]	3.48725349e+00	2.22682946e-02	1.97392185e-02
CaO[S]	6.61375661e-02	4.22329724e-04	6.80678509e-04
Cr203[S]	2.40500241e-02	1.53574445e-04	9.13231864e-05
Fe203[S1]	6.07263107e+01	3.87775474e-01	2.19475361e-01
Fe304[S1]	0.00000000e+00	0.0000000e+00	0.00000000e+00
FeO[S]	8.38744589e+01	5.35590878e-01	6.73780610e-01
K20[S]	1.20250120e-02	7.67872226e-05	7.36774791e-05
Mg0[S]	3.12650313e+00	1.99646779e-02	4.47701043e-02
MnO[S]	1.62337662e+00	1.03662751e-02	1.32076460e-02
Na20[S1]	2.10437710e-02	1.34377640e-04	1.95957154e-04
P4010[S]	3.00625301e-03	1.91968057e-05	6.11165160e-06
SiO2[S1]	2.55531506e+00	1.63172848e-02	2.45451193e-02
TiO2[S1]	0.00000000e+00	0.00000000e+00	0.00000000e+00
V205[S]	1.08225108e+00	6.91085003e-03	3.43419366e-03
MaterialPackage			
========= Material	Ilmenite	=========	
Mass	1.43398268e+02	kg	
7	1.79549029e+00		
AIIIOUIIC			
	1.00000000e+00	atm	
Pressure	1.00000000e+00 2.50000000e+01		
Pressure Temperature		°C	
Amount Pressure Temperature Enthalpy Compound Details	2.50000000e+01	°C	
Pressure Temperature Enthalpy Compound Details	2.50000000e+01	°C	 Mole Fraction
Pressure Temperature Enthalpy Compound Details Formula	2.50000000e+01 -4.71191509e+02	°C kWh Mass Fraction	
Pressure Temperature Enthalpy Compound Details Formula Al203[S1]	2.50000000e+01 -4.71191509e+02 	°C kWh Mass Fraction	Mole Fraction 0.000000000e+00 0.000000000e+00
Pressure Temperature Enthalpy Compound Details Formula Al203[S1] Ca0[S]	2.50000000e+01 -4.71191509e+02 Mass 0.00000000e+00	°C kWh Mass Fraction 0.00000000e+00	0.00000000e+00
Pressure Temperature Enthalpy Compound Details Formula Al203[S1] Ca0[S] Cr203[S]	2.50000000e+01 -4.71191509e+02 Mass 0.00000000e+00 0.00000000e+00	°C kWh Mass Fraction 0.000000000e+00 0.000000000e+00	0.00000000e+00
Pressure Temperature Enthalpy Compound Details Formula Al203[S1] Ca0[S] Cr203[S] Fe203[S1]	2.50000000e+01 -4.71191509e+02 Mass 0.00000000e+00 0.00000000e+00 0.00000000e+00	°C kWh Mass Fraction 0.00000000e+00 0.00000000e+00 0.00000000e+00	0.00000000e+00 0.00000000e+00 0.00000000e+00
Pressure Temperature Enthalpy 	2.50000000e+01 -4.71191509e+02 Mass 0.000000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00	°C kWh Mass Fraction 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00	0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
Pressure Temperature Enthalpy Compound Details Formula Al203[S1] Ca0[S] Cr203[S] Fe203[S1] Fe304[S1]	2.50000000e+01 -4.71191509e+02 Mass 0.000000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00	°C kWh Mass Fraction 0.000000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00	0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00

MnO[S]	0.00000000e+00	0.00000000e+00	0.00000000e+00
Na20[S1]	0.00000000e+00	0.00000000e+00	0.00000000e+00
P4010[S]	0.00000000e+00	0.00000000e+00	0.00000000e+00
SiO2[S1]	0.00000000e+00	0.00000000e+00	0.00000000e+00
TiO2[S1]	1.43398268e+02	1.00000000e+00	1.00000000e+00
V205[S]	0.00000000e+00	0.00000000e+00	0.00000000e+00
=======================================			

Extract Material from a Package - Compound Mass

We may not want to extract all the mass of a specific compound. In this case we can specify the mass to extract.

```
from auxi.modelling.process.materials.thermo import Material

ilmenite = Material("Ilmenite", "./materials/ilmenite.txt")

reductant = Material("Reductant", "./materials/reductant.txt")

mix = Material("Mix", "./materials/mix.txt")

ilma_package = ilmenite.create_package("IlmeniteA", 300.0, 1.0, 25.

o)

new_package = ilma_package.extract(("TiO2[S1]", 110.0))

print(ilma_package)

print(new_package)
```

The existing package is modified appropriately and a new package containing only the specified mass of the required compound is produced.

```
______
MaterialPackage
______
Material
               Ilmenite
Mass
               1.90000000e+02 kg
Amount
                2.15085961e+00 kmol
               1.00000000e+00 atm
Pressure
               2.50000000e+01 °C
Temperature
               -3.26363778e+02 kWh
Enthalpy
Compound Details
Formula
              Mass
                           Mass Fraction Mole Fraction
Al203[S1]
               3.48725349e+00 1.83539657e-02 1.59014304e-02
              6.61375661e-02 3.48092453e-04 5.48337915e-04
CaO[S]
               2.40500241e-02 1.26579074e-04 7.35677195e-05
Cr203[S]
Fe203[S1]
               6.07263107e+01 3.19612162e-01 1.76803968e-01
Fe304[S1]
               0.00000000e+00 0.0000000e+00 0.0000000e+00
               8.38744589e+01 4.41444520e-01 5.42781136e-01
FeO[S]
               1.20250120e-02 6.32895370e-05 5.93527704e-05
K20[S]
```

Mg0[S]	3.12650313e+00	1.64552796e-02	3.60656982e-02
MnO[S]	1.62337662e+00	8.54408749e-03	1.06397557e-02
Na20[S1]	2.10437710e-02	1.10756690e-04	1.57858278e-04
P4010[S]	3.00625301e-03	1.58223842e-05	4.92339666e-06
SiO2[S1]	2.55531506e+00	1.34490266e-02	1.97729462e-02
TiO2[S1]	3.33982684e+01	1.75780360e-01	1.94424522e-01
V205[S]	1.08225108e+00	5.69605833e-03 	2.76650220e-03
======================================	==========		========
==========	=======================================		=========
Material	Ilmenite		
Mass	1.10000000e+02	_	
Amount	1.37731044e+00		
Pressure	1.00000000e+00		
Temperature	2.50000000e+01		
Enthalpy	-3.61448340e+02	kWh	
Compound Details			
Compound Details Formula	Mass	Mass Fraction	Mole Fraction
-	Mass 0.000000000000000000000000000000	Mass Fraction 0.00000000e+00	Mole Fraction 0.000000000e+00
Formula			
Formula Al203[S1]	0.00000000e+00	0.00000000e+00	0.00000000e+00
Formula Al203[S1] Ca0[S] Cr203[S]	0.00000000e+00 0.00000000e+00	0.00000000e+00 0.00000000e+00	0.00000000e+00 0.00000000e+00
Formula Al203[S1] Ca0[S] Cr203[S] Fe203[S1]	0.00000000e+00 0.00000000e+00 0.00000000e+00	0.00000000e+00 0.00000000e+00 0.00000000e+00	0.00000000e+00 0.00000000e+00 0.00000000e+00
Formula Al203[S1] Ca0[S] Cr203[S] Fe203[S1] Fe304[S1]	0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00	0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00	0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
Formula	0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00	0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00	0.00000000e+00 0.000000000e+00 0.000000000e+00 0.00000000e+00 0.000000000e+00
Formula	0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00	0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00	0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
Formula	0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00	0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00	0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
Formula Al203[S1] CaO[S]	0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00	0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00	0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
Formula	0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00	0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00	0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
Formula	0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00	0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00	0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
Formula	0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00	0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00	0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00

Extract Material from a Package - Material

We may need to extract all the compounds that appear in a specific material into a new package.

```
from auxi.modelling.process.materials.thermo import Material

ilmenite = Material("Ilmenite", "./materials/ilmenite.txt")

reductant = Material("Reductant", "./materials/reductant.txt")

mix = Material("Mix", "./materials/mix.txt")

ilma_package = ilmenite.create_package("IlmeniteA", 300.0, 1.0, 25.
```

```
new_package = ilma_package.extract(reductant)
print(ilma_package)
print(new_package)
```

The existing package loses all the masses of components that appear in the specified material. The new package contains these masses and have the same temperature and pressure as the original material.

Material	T1		
Mass	Ilmenite 2.30038480e+02	ka	
Amount	2.99240730e+00	_	
Pressure	1.00000000e+00		
Temperature	2.500000000e+01		
Enthalpy	-5.62518853e+02		
Compound Details			
Formula	Mass	Mass Fraction	Mole Fraction
Al203[S1]	0.00000000e+00	0.00000000e+00	0.0000000e+00
CaO[S]	0.00000000e+00	0.00000000e+00	0.00000000e+00
Cr203[S]	2.40500241e-02	1.04547831e-04	5.28784420e-05
Fe203[S1]	0.00000000e+00	0.00000000e+00	0.0000000e+00
Fe304[S1]	0.00000000e+00	0.00000000e+00	0.00000000e+00
FeO[S]	8.38744589e+01	3.64610559e-01	3.90136068e-01
K20[S]	1.20250120e-02	5.22739153e-05	4.26611298e-05
Mg0[S]	0.00000000e+00	0.00000000e+00	0.00000000e+00
MnO[S]	1.62337662e+00	7.05697857e-03	7.64756215e-03
Na20[S1]	2.10437710e-02	9.14793518e-05	1.13464164e-04
P4010[S]	3.00625301e-03	1.30684788e-05	3.53880134e-06
SiO2[S1]	0.00000000e+00	0.00000000e+00	0.00000000e+00
TiO2[S1]	1.43398268e+02	6.23366440e-01	6.00015342e-01
V205[S] ========	1.08225108e+00	4.70465238e-03	1.98848527e-03
======================================	:=========		========
=======================================			========
Material	Reductant		
Mass	6.99615200e+01	-	
Amount	5.35762740e-01		
Pressure	1.00000000e+00		
Temperature	2.50000000e+01		
Enthalpy	-1.25293265e+02	kWh	
Compound Details			
Formula	Mass	Mass Fraction	Mole Fraction

```
C[S1]
                  0.0000000e+00 0.0000000e+00 0.0000000e+00
                  0.0000000e+00 0.0000000e+00 0.0000000e+00
H2[G]
02 [G]
                 0.0000000e+00 0.0000000e+00 0.0000000e+00
N2[G]
                 0.0000000e+00 0.0000000e+00 0.0000000e+00
                 0.0000000e+00 0.0000000e+00 0.0000000e+00
S[S1]
                 3.48725349e+00 4.98453077e-02 6.38374820e-02
A1203[S1]
                  6.61375661e-02 9.45342042e-04 2.20134358e-03
CaO[S]
Fe203[S1]
                 6.07263107e+01 8.67995875e-01 7.09792759e-01
Mg0[S]
                  3.12650313e+00 4.46888965e-02 1.44788444e-01
                  2.55531506e+00 3.65245789e-02 7.93799718e-02
SiO2[S1]
```

Multiplying a Package by a Scalar

It may sometimes be useful to multiply a package by a scalar.

```
from auxi.modelling.process.materials.thermo import Material

ilmenite = Material("Ilmenite", "./materials/ilmenite.txt")

reductant = Material("Reductant", "./materials/reductant.txt")

mix = Material("Mix", "./materials/mix.txt")

ilma_package = ilmenite.create_package("IlmeniteA", 300.0, 1.0, 25.

o)

ilma_package *= 2.0

print(ilma_package)
```

This doubles the package mass and enthalpy. Temperature, pressure and composition remain the same, since these are intensive properties.

```
MaterialPackage
______
Material
               Ilmenite
Mass
                6.00000000e+02 kg
Amount
                7.05634009e+00 kmol
Pressure
                1.00000000e+00 atm
Temperature
                2.50000000e+01 °C
Enthalpy
                -1.37562424e+03 kWh
Compound Details
                            Mass Fraction Mole Fraction
Formula
               Mass
A1203[S1]
               6.97450697e+00 1.16241783e-02 9.69390473e-03
               1.32275132e-01 2.20458554e-04 3.34280337e-04
CaO[S]
               4.81000481e-02 8.01667468e-05 4.48486990e-05
Cr203[S]
              1.21452621e+02 2.02421036e-01 1.07784066e-01
Fe203[S1]
```

Fe304[S1]	0.00000000e+00	0.0000000e+00	0.00000000e+00
FeO[S]	1.67748918e+02	2.79581530e-01	3.30892788e-01
K20[S]	2.40500241e-02	4.00833734e-05	3.61829148e-05
Mg0[S]	6.25300625e+00	1.04216771e-02	2.19865404e-02
MnO[S]	3.24675325e+00	5.41125541e-03	6.48625791e-03
Na20[S1]	4.20875421e-02	7.01459035e-05	9.62343053e-05
P4010[S]	6.01250601e-03	1.00208434e-05	3.00142421e-06
SiO2[S1]	5.11063011e+00	8.51771685e-03	1.20540764e-02
TiO2[S1]	2.86796537e+02	4.77994228e-01	5.08901291e-01
V2O5[S]	2.16450216e+00	3.60750361e-03	1.68652807e-03
=======================================			=========

Setting Package Temperature

Using the "T" property of a MaterialPackage object, it is easy to set the temperature of a package to a new value.

```
from auxi.modelling.process.materials.thermo import Material

ilmenite = Material("Ilmenite", "./materials/ilmenite.txt")

reductant = Material("Reductant", "./materials/reductant.txt")

mix = Material("Mix", "./materials/mix.txt")

ilma_package = ilmenite.create_package("IlmeniteA", 300.0, 1.0, 25.

o)

ilma_package.T = 1000.0

print(ilma_package)
```

This results in the temperature to be updated, as well as the package's enthalpy.

```
______
MaterialPackage
______
Material
              Ilmenite
               3.00000000e+02 kg
Mass
Amount
               3.52817004e+00 kmol
Pressure
              1.00000000e+00 atm
              1.00000000e+03 °C
Temperature
Enthalpy
              -6.18986580e+02 kWh
Compound Details
                    Mass Fraction Mole Fraction
Formula
              Mass
Al203[S1]
              3.48725349e+00 1.16241783e-02 9.69390473e-03
Ca0[S]
              6.61375661e-02 2.20458554e-04 3.34280337e-04
              2.40500241e-02 8.01667468e-05 4.48486990e-05
Cr203[S]
              6.07263107e+01 2.02421036e-01 1.07784066e-01
Fe203[S1]
             0.00000000e+00 0.0000000e+00 0.0000000e+00
Fe304[S1]
```

```
FeO[S]
                8.38744589e+01 2.79581530e-01 3.30892788e-01
K20[S]
                1.20250120e-02 4.00833734e-05 3.61829148e-05
                 3.12650313e+00 1.04216771e-02 2.19865404e-02
Mg0[S]
MnO[S]
                1.62337662e+00 5.41125541e-03 6.48625791e-03
Na20[S1]
                2.10437710e-02 7.01459035e-05 9.62343053e-05
                3.00625301e-03 1.00208434e-05 3.00142421e-06
P4010[S]
                2.55531506e+00 8.51771685e-03 1.20540764e-02
SiO2[S1]
                 1.43398268e+02 4.77994228e-01 5.08901291e-01
TiO2[S1]
V205[S]
                1.08225108e+00 3.60750361e-03 1.68652807e-03
______
```

Setting Package Enthalpy

We can use the "H" property of a MaterialPackage object to add or subtract enthalpy, or to set it to a new value. This is very useful when building an energy balance.

```
from auxi.modelling.process.materials.thermo import Material

ilmenite = Material("Ilmenite", "./materials/ilmenite.txt")

reductant = Material("Reductant", "./materials/reductant.txt")

mix = Material("Mix", "./materials/mix.txt")

ilma_package = ilmenite.create_package("IlmeniteA", 300.0, 1.0, 25.

o)

ilma_package.H = ilma_package.H + 1.0

print(ilma_package)
```

This updates the package's enthalpy and automatically re-calculates its temperature.

```
______
MaterialPackage
______
Material
             Ilmenite
Mass
              3.00000000e+02 kg
              3.52817004e+00 kmol
Amount
Pressure
              1.00000000e+00 atm
             4.22166385e+01 °C
Temperature
              -6.86812118e+02 kWh
Enthalpy
Compound Details
Formula
                         Mass Fraction Mole Fraction
_____
              3.48725349e+00 1.16241783e-02 9.69390473e-03
Al203[S1]
              6.61375661e-02 2.20458554e-04 3.34280337e-04
CaO[S]
Cr203[S]
             2.40500241e-02 8.01667468e-05 4.48486990e-05
Fe203[S1]
             6.07263107e+01 2.02421036e-01 1.07784066e-01
             0.00000000e+00 0.0000000e+00 0.0000000e+00
Fe304[S1]
             8.38744589e+01 2.79581530e-01 3.30892788e-01
FeO[S]
```

K20[S]	1.20250120e-02	4.00833734e-05	3.61829148e-05
MgO[S]	3.12650313e+00	1.04216771e-02	2.19865404e-02
MnO[S]	1.62337662e+00	5.41125541e-03	6.48625791e-03
Na20[S1]	2.10437710e-02	7.01459035e-05	9.62343053e-05
P4010[S]	3.00625301e-03	1.00208434e-05	3.00142421e-06
SiO2[S1]	2.55531506e+00	8.51771685e-03	1.20540764e-02
TiO2[S1]	1.43398268e+02	4.77994228e-01	5.08901291e-01
V205[S]	1.08225108e+00	3.60750361e-03	1.68652807e-03
=========			==========

3.2.2 Business Modelling

The purpose of this section is to explain a number of concepts and demonstrate the use of the Entity, Component, Activity classes in the auxi.modelling.business module.

Basic Activity

A basic activity periodically create a transaction between two specified accounts.

To create an basic activity, import the 'BasicActivity' and the create a 'BasicActivity'

CHAPTER

FOUR

AUXI REFERENCE

4.1 auxi package

4.1.1 Subpackages

auxi.core package

Submodules

auxi.core.bibliography module

auxi.core.bibliography_test module

auxi.core.helpers module

This module contains helper functions to ease various tasks.

```
auxi.core.helpers.get_date ( date)
```

Get the date from a value that could be a date object or a string.

Parameters date – The date object or string.

Returns The date object.

Calculate a path relative to the specified module file.

Parameters module_file_path – The file path to the module.

auxi.core.helpers_test module

This module contains code used to test core helpers classes.

```
class auxi.core.helpers_test. HelpersUnitTester ( methodName='runTest')
     Bases: unittest.case.TestCase
     The unit tester for the helpers functions being tested.
     test_get_date ( )
     test_get_path_relative_to_module ()
auxi.core.objects module
class auxi.core.objects. NamedObject ( name, description=None)
     Bases: auxi.core.objects.Object
     Base class for all auxi classes requiring a name and description.
         Parameters
               • name – the object's name
               • description – the object's description
     name
class auxi.core.objects. Object
     Bases: object
     Base class for all auxi classes.
     read (path)
     write ( path)
auxi.core.objects_test module
This module contains code used to test core object classes.
class auxi.core.objects_test. NamedObjectUnitTester ( methodName='runTest')
     Bases: unittest.case.TestCase
     The unit tester for the class being tested.
     setUp ( )
     tearDown ()
     test str ()
         Test whether the __str__ method successfully generates a json string representation
         of the object.
     test_constructor()
         Test whether the constructor successfully initialises the object.
class auxi.core.objects_test. ObjectUnitTester ( methodName='runTest')
     Bases: unittest.case.TestCase
```

The unit tester for the class being tested.

```
setUp ( )
tearDown ( )
test___str__ ( )
    Test whether the __str__ method successfully generates a json string representation
    of the object.
```

auxi.core.reporting module

This module provides classes to create reports.

```
class auxi.core.reporting. Report ( data_source, output_path=None)
    Bases: auxi.core.objects.Object

Base class for all auxi reports.

render ( format=<ReportFormat.printout: (1, )>)
    Render the report in the specified format
```

Parameters format – The format. The default format is to print the report to the console.

Returns If the format was set to 'string' then a string representation of the report is returned.

```
class auxi.core.reporting. ReportFormat
    Bases: enum.Enum
    Represents the format the report should be outputted as.
```

```
csv = <ReportFormat.csv: (4,)>
latex = <ReportFormat.latex: (2,)>
matplotlib = <ReportFormat.matplotlib: (6,)>
png = <ReportFormat.png: 7>
printout = <ReportFormat.printout: (1,)>
string = <ReportFormat.string: (5,)>
txt = <ReportFormat.txt: (3,)>
```

auxi.core.reporting test module

This module contains code used to test core reporting classes.

```
\begin{tabular}{ll} \textbf{class} & \texttt{auxi.core.reporting\_test.} & \textbf{ReportingUnitTester} & (\textit{methodName} = \textit{`runTest'}) \\ & \textbf{Bases:} & \texttt{unittest.case.TestCase} \\ \end{tabular}
```

The unit tester for the class being tested.

```
setUp ()
     tearDown ()
     test_render_with_file ()
          Test whether a value is returned when a report is rendered in a string format. And
          wether any of the other formats throw an error.
     test_render_without_file ()
          Test whether a value is returned when a report is rendered in a string format. And
          wether any of the other formats throw an error.
auxi.core.time module
This module provides classes to manage time.
class auxi.core.time. Clock (name,
                                                              description=None,
                                   start datetime=datetime.datetime(1, 1, 1, 0, 0),
                                   timestep_period_duration=<TimePeriod.month:
                                   7>, timestep_period_count=1)
     Bases: auxi.core.objects.NamedObject
     Represents a clock that provices functions to manage a ticking clock based on a time
     period as well as retreive the current tick's date since the start date.
     get_datetime ( )
          Get the clock's current datetime.
              Returns The datetime.
     get_datetime_at_period_ix ( ix)
          Get the datetime at a given period.
              Parameters period – The index of the period.
              Returns The datetime.
     reset ()
          Resets the clock's timestep index to '0'.
     tick ()
          Increment the clock's timestep index.
class auxi.core.time. TimePeriod
     Bases: enum. Enum
     Represents a period in time.
     day = <TimePeriod.day: 5>
     hour = <TimePeriod.hour: 4>
     millisecond = <TimePeriod.millisecond: 1>
     minute = <TimePeriod.minute: 3>
```

month = <TimePeriod.month: 7>

```
second = <TimePeriod.second: 2>
week = <TimePeriod.week: 6>
year = <TimePeriod.year: 8>
```

auxi.core.time test module

This module provides testing code for the auxi.core.time module.

```
class auxi.core.time_test. ClockUnitTester ( methodName='runTest')
    Bases: unittest.case.TestCase
    Tester for the auxi.core.time.Clock class.
    setUp ()
    test_constructor ()
    test_get_datetime ()
    test_get_datetime_at_period_ix ()
    test_reset ()
    test_tick ()
```

Module contents

This package contains foundational infrastructure classes and functions that are used throughout the rest of auxi.

auxi.modelling package

Subpackages

auxi.modelling.business package

Submodules

auxi.modelling.business.basic module

This module provides class and functions for basic business activities.

Bases: auxi.modelling.business.structure.Activity

An activity class that provides the most basic activity functionality: periodically create a transaction between two specified accounts.

Parameters

- name The name.
- dt_account The debit account.
- cr_account The credit account.
- amount The amount of an activity.
- **start** The datetime the activity should be started.
- end The datetime the activity should be run until.
- **interval** The interval of the activity.
- **description** The description.

get_referenced_accounts ()

Retrieve the general ledger accounts referenced in this instance.

Returns The referenced accounts.

```
run ( clock, generalLedger)
```

Execute the activity at the current clock cycle.

Parameters

- **clock** The clock containing the current execution time and period information.
- **generalLedger** The general ledger into which to create the transactions.

```
class auxi.modelling.business.basic. BasicLoanActivity (name,
                                                                       bank_account,
                                                                       loan_account,
                                                                       inter-
                                                                       est_account,
                                                                       amount=0,
                                                                       inter-
                                                                       est rate=0.0,
                                                                       start=datetime.datetime(1,
                                                                       1, 1, 0,
                                                                       0).
                                                                       dura-
                                                                       tion=60.
                                                                       inter-
                                                                       val=1,
                                                                       descrip-
                                                                       tion=None)
```

Bases: auxi.modelling.business.structure.Activity

An activity class that provides the most basic activity functionality for a loan: Creates a loan transaction and periodically create transactions to consider the interest and to pay the interest.

Parameters

- name The name.
- **description** The description.
- bank_account The asset account that is increased.
- **loan_account** The liability account that is decreased.
- interest_account The expense account the interest is added to.
- amount The loan amount. The default amount is 0
- **interest_rate** The interest rate as a fraction of the whole (e.g. 0.15 = 15%). The default value is 0.0
- **start** The datetime the activity should be started.
- duration The duration of the loan in months.
- **interval** The interval of the activity.

amount

get_referenced_accounts ()

Retrieve the general ledger accounts referenced in this instance.

Returns The referenced accounts.

interest rate

```
prepare_to_run ( clock, period_count)
```

Prepare the activity for execution.

Parameters

- **clock** The clock containing the execution start time and execution period information.
- **period_count** The total amount of periods this activity will be requested to be run for.

```
run ( clock, generalLedger)
```

Execute the activity at the current clock cycle.

Parameters

- **clock** The clock containing the current execution time and period information.
- **generalLedger** The general ledger into which to create the transactions.

auxi.modelling.business.basic_test module

This module provides testing code for the auxi.modelling.business.basic module.

Tester for the auxi.modelling.business.basic.BasicActivity class.

```
setUp ()
```

```
test__meet_exection_criteria ( )
```

Test that the activity only meets the execution criteria when it's amount is greater than 0.

```
test_constructor ( )
```

```
test_get_referenced_accounts ()
```

Test that the activity run method get_referenced_accounts accounts matches the debit and credit accounts self.object was initialised with.

```
test_run ( )
```

Test that the activity run method creates a transaction with an amount of 5000.

Tester for the auxi.modelling.business.basic.BasicLoanActivity class.

```
setUp ( )
```

```
test__meet_exection_criteria ()
```

Test that the activity only meets the execution criteria when it's amount is greater than 0 and its duration is greater than 0.

```
test_constructor ( )
```

test_get_referenced_accounts ()

Test that the activity run method get_referenced_accounts accounts matches the debit and credit accounts self.object was initialised with.

```
test_run_first_month ( )
```

Test that the activity run method creates a transaction with an amount of 180000 on the first month. No other transactions

```
test_run_last_month ( )
```

Test that the activity run method creates settled the loan on the loans' last month and that the transactions

```
test_run_third_month()
```

Test that the activity run method accrued the interest correctly.

auxi.modelling.business.models module

This module provides classes to work with business models.

Represents an time based model class. An instance of this class is by default configured to run only once, thus functioning as a steady state model. The instance's time based parameters must be configured for it to function as a time based model.

Parameters

- name The name.
- **description** The description.
- **start_datetime** The start datetime of the model.
- **period_duration** The duration of the model's time period. e.g. month, day etc.
- **period_count** The number of periods to execute the model for.

```
create_entity ( name, gl_structure, description=None)
```

Create an entity and add it to the model.

Parameters

- name The entity name.
- **gl_structure** The entity's general ledger structure.
- **description** The entity description.

Returns The created entity.

name

```
prepare_to_run ( )
```

Prepare the model for execution.

```
remove_entity ( name)
```

Remove an entity from the model.

Parameters name – The name of the entity to remove.

```
run ()
```

Execute the model.

auxi.modelling.business.models_test module

This module provides testing code for the classes in the auxi.modelling.business.models module.

Tester for the auxi.modelling.business.models.TimeBasedModel class.

setUp ()

```
test_constructor ( )
```

test_create_entity()

```
test_prepare_to_run ( )
```

Test that the model's entities prepare_to_run is called when the model's prepare_to_run is called.

```
test_remove_entity()
```

```
test_run ()
```

Test that the model's entities run is called when the model's run is called. Also test that the model was run for the expected timeperiod.

```
test_set_name ( )
```

Test wether the name changes when it is set, that the model's path is updated and that the model's children's paths are updated correctly.

auxi.modelling.business.structure module

This module provides an classes used to create a business structure.

Bases: auxi.core.objects.NamedObject

Represents an activity base class. An activity will typically represent a transaction activity in a business.

Parameters

- name The name.
- **description** The description.
- **start** The datetime the activity should be started.
- end The datetime the activity should be run until.
- **interval** The interval of the activity.

```
get_referenced_accouts ()
```

Retrieve the general ledger accounts referenced in this instance.

Returns The referenced accounts.

name

```
{\tt prepare\_to\_run} \ (\ clock, period\_count)
```

Prepare the activity for execution.

Parameters

- **clock** The clock containing the execution start time and execution period information.
- **period_count** The total amount of periods this activity will be requested to be run for.

```
set_parent_path (value)
```

Set the parent path and the path from the new parent path.

Parameters value – The path to the object's parent

```
class auxi.modelling.business.structure. Component (name, gl, description=None) Bases: auxi.core.objects.NamedObject
```

Represents an component class. A component class that represents a component of an entity. A component has business activities

Parameters

- name The name.
- **description** The description.

add_activity (activity)

Add an activity to the component.

Parameters activity – The activity.

```
create_component ( name, description=None)
```

Create a sub component in the business component.

Parameters

- name The new component's name.
- **description** The new component's description.

Returns The created component.

```
get_activity ( name)
```

Retrieve an activity given its name.

Parameters name – The name of the activity.

Returns The activity.

```
get_component ( name)
```

Retrieve a child component given its name.

Parameters name – The name of the component.

Returns The component.

name

```
prepare_to_run ( clock, period_count)
```

Prepare the component for execution.

Parameters

- **clock** The clock containing the execution start time and execution period information.
- **period_count** The total amount of periods this activity will be requested to be run for.

```
remove_component ( name)
```

Remove a sub component from the component.

Parameters name – The name of the component to remove.

```
run ( clock, generalLedger)
```

Execute the component at the current clock cycle.

Parameters

- **clock** The clock containing the current execution time and period information.
- generalLedger The general ledger into which to create the transactions.

```
set_parent_path ( value)
```

Set the parent path and the path from the new parent path.

Parameters value – The path to the object's parent.

```
class auxi.modelling.business.structure. Entity ( name, gl\_structure, description=None)
```

Bases: auxi.core.objects.NamedObject

Represents an entity class. An entity consists of business components e.g. Sales department. It executes its components and performs financial year end transcations.

Parameters

- name The name.
- **gl_structure** The general ledger structure the entity's general ledger will be initialized with.
- **description** The description.
- **period_count** The number of periods the entity should be run for.

```
create_component ( name, description=None)
```

Create a component in the business entity.

Parameters

- name The component's name.
- **description** The component's description.

Returns The created component.

name

```
prepare_to_run ( clock, period_count)
```

Prepare the entity for execution.

Parameters

- **clock** The clock containing the execution start time and execution period information.
- **period_count** The total amount of periods this activity will be requested to be run for.

```
remove_component ( name)
```

Remove a component from the entity.

```
Parameters name – The name of the component to remove.
```

```
run ( clock)
```

Execute the entity at the current clock cycle.

Parameters clock – The clock containing the current execution time and period information.

```
set_parent_path (value)
```

Set the parent path and the path from the new parent path.

Parameters value – The path to the object's parent

auxi.modelling.business.structure_test module

This module provides testing code for the auxi.modelling.business.activity module.

```
@author: Ex Mente Technologies (Pty) Ltd
```

Tester for the auxi.modelling.business.structure.Activity class.

```
setUp ( )

test__meet_exection_criteria ( )

test_constructor ( )

test_get_path ( )

test_get_referenced_accounts ( )

test_prepare_to_run ( )

test_set_name ( )

Test wether the name changes when it is set and that the activity's path is updated correctly.
```

test_set_parent_path ()

Tester for the auxi.modelling.business.structure.Component class.

```
setUp ()
test_add_activity_invalid_account_name ()
test_add_activity_valid_account_name ()
test_constructor ()
test_create_component ()
test_get_activity_exists ()
```

```
test_get_activity_not_exists()
     test get component exists ()
    test_get_component_not_exists()
     test_prepare_to_run ( )
         Test that the component run's its activities' prepare_to_run methods.
    test_remove_component ( )
     test run ()
         Test that the component runs its activities.
     test_set_name ( )
         Test wether the name changes when it is set, that the component's name changes
         and that the component's children's paths are updated correctly.
     test_set_parent_path ( )
class auxi.modelling.business.structure_test. EntityUnitTester ( methodName='ru
     Bases: unittest.case.TestCase
     Tester for the auxi.modelling.entity.component class.
     setUp ()
     test__perform_year_end_gross_profit ( )
         Test that the year end gross profit account is closed correctly.
     test__perform_year_end_gross_profit_and_income_summary ( )
         Test that the year end gross profit and income summary accounts are summed up
         correctly.
     test__perform_year_end_income_summary ( )
         Test that the year end income summary account is closed correctly.
     test__perform_year_end_income_tax ( )
     test__perform_year_end_procedure ( )
         Test that all the year end accounts has been closed of correctly.
     test__perform_year_end_retained_earnings()
         Test that the year end retained earnings account is closed correctly.
     test_constructor()
    test_create_component ( )
    test_getitem_exists()
    test_getitem_not_exists()
     test_prepare_to_run ( )
         Test that the entity run's its component' prepare_to_run methods.
     test_remove_component ( )
```

```
test run ()
```

Test that the entity runs its components.

```
test_set_name ( )
```

Test wether the name changes when it is set, that the entity's name changes and that the component's children's paths are updated correctly.

```
test_set_parent_path ( )
```

Module contents

auxi.modelling.business Package

The purpose of this package is to provide common classes and functions for the auxi.modelling.business namespace

auxi.modelling.financial package

Submodules

auxi.modelling.financial.des module

This module provides classes representing the accounting double entry system.

```
auxi.modelling.financial.des. AT
     alias of Account Type
class auxi.modelling.financial.des. AccountType
     Bases: enum. Enum
    Represents the type of general ledger account.
     asset = <AccountType.asset: 1>
     equity = <AccountType.equity: 2>
     expense = <AccountType.expense: 3>
     liability = <AccountType.liability: 4>
     revenue = <AccountType.revenue: 5>
class auxi.modelling.financial.des. GeneralLedger (name,
                                                                descrip-
                                                          ture,
                                                          tion=None)
```

Bases: auxi.core.objects.NamedObject Represents the account structure of a general ledger.

Parameters

- name The name.
- **structure** The general ledger structure.

struc-

• description – The description.

balance_sheet (end=datetime.datetime(9999, 12, 31, 23, 59, 59, 999999), format=<ReportFormat.printout: (1,)>, out-put_path=None)

Generate a transaction list report.

Parameters

- **end** The end date to generate the report for.
- **format** The format of the report.
- output_path The path to the file the report is written to. If None, then the report is not written to a file.

Returns The generated report.

Parameters

- name The transaction's name.
- **description** The transaction's description.
- tx_date The date of the transaction.
- **cr_account** The transaction's credit account's name.
- dt_account The transaction's debit account's name.
- **source** The name of source the transaction originated from.
- amount The transaction amount.

Returns The created transaction.

income_statement (start=datetime.datetime(1, 1, 1, 0, 0), end=datetime.datetime(9999, 12, 31, 23, 59, 59, 999999), $format=<ReportFormat.printout: (1,)>, component_path='`, output_path=None')$ Generate a transaction list report.

Parameters

- **start** The start date to generate the report for.
- **end** The end date to generate the report for.
- **format** The format of the report.
- **component_path** The path of the component to filter the report's transactions by.
- output_path The path to the file the report is written to. If None, then the report is not written to a file.

Returns The generated report.

```
transaction_list ( start=datetime.datetime(1, 1, 1, 0, 0), end=datetime.datetime(9999, 12, 31, 23, 59, 59, 999999), format=<ReportFormat.printout: (1, )>, component\_path='`, output\_path=None') Generate a transaction list report.
```

Parameters

- **start** The start date to generate the report for.
- **end** The end date to generate the report for.
- **format** The format of the report.
- **component_path** The path of the component to filter the report's transactions by.
- output_path The path to the file the report is written to. If None, then the report is not written to a file.

Returns The generated report.

Bases: auxi.core.objects.NamedObject

Represents an account of a general ledger.

Parameters

- name The name.
- **description** The description.
- number The number.
- account_type The type of account.

create_account (name, number=None, description=None)
Create a sub account in the account.

Parameters

- name The account name.
- **description** The account description.
- **number** The account number.

Returns The created account.

get_child_account (account_name)

Retrieves a child account. This could be a descendant nested at any level.

Parameters account_name – The name of the account to retrieve.

Returns The child account, if found, else None.

name

```
remove_account ( name)
```

Remove an account from the account's sub accounts.

Parameters name – The name of the account to remove.

```
set_parent_path (value)
```

Set the parent path and the path from the new parent path.

Parameters value – The path to the object's parent

```
class auxi.modelling.financial.des. GeneralLedgerStructure ( name,
```

de-

scrip-

tion=None)

Bases: auxi.core.objects.NamedObject

The account structure of a general ledger.

Parameters

- name The name.
- **description** The description.

```
get_account ( account_name)
```

Retrieves an account from the general ledger structure given the account name.

Parameters account_name – The account name.

Returns The requested account, if found, else None.

```
get_account_decendants ( account)
```

Retrieves an account's decendants from the general ledger structure given the account name.

Parameters account_name – The account name.

Returns The decendants of the account.

```
report (format=<ReportFormat.printout: (1, )>, output_path=None)
```

Returns a report of this class.

Parameters

- **format** The format of the report.
- output_path The path to the file the report is written to. If None, then the report is not written to a file.

Returns The decendants of the account.

```
validate_account_names ( names)
```

Validates wether the accounts in a list of account names exists.

Parameters names – The names of the accounts.

Returns The decendants of the account.

Bases: auxi.core.objects.NamedObject

Represents a financial transaction between two general ledger accounts.

Parameters

- name The name.
- **description** The description.
- tx date The transaction's date.
- dt_account The account to debit.
- **cr_account** The account to credit.
- **source** The source that created the transaction.
- amount The transaction's amount.
- is_closing_dt_account Specifies wether this is a closing debit account.
- is_closing_cr_account Specifies wether this is a closing credit account.

Bases: auxi.core.objects.NamedObject

Represents a template for how a transaction is to be created.

Parameters

- name The name of the transaction.
- **description** The description of the transaction.

- dt_account The account to debit.
- cr account The account to credit.

auxi.modelling.financial.des_test module

```
This module provides testing code for the auxi.modelling.financial.des module.
```

```
class auxi.modelling.financial.des_test. GeneralLedgerAccountUnitTester ( meth
    Bases: unittest.case.TestCase
    Tester for the auxi.modelling.financial.des.GeneralLedgerAccount class.
    setUp ( )
    test_constructor ( )
    test_create_account ( )
    test_get_child_account ()
    test_remove_account ( )
    test_set_parent_path ( )
class auxi.modelling.financial.des_test. GeneralLedgerStructureUnitTester ( n
    Bases: unittest.case.TestCase
    Tester for the auxi.modelling.financial.des.generalledgerstructure class.
    setUp ( )
    test_constructor()
         Test the the variables has been initialised and that the default accounts has been
         created.
    test_get_account ()
    test_get_account_decendants ( )
    test_report ( )
    test_validate_account_name_invalid ( )
    test_validate_account_name_valid ( )
class auxi.modelling.financial.des_test. GeneralLedgerUnitTester ( methodName=
    Bases: unittest.case.TestCase
    Tester for the auxi.modelling.financial.des.GeneralLedger class.
    setUp ( )
    test_constructor ( )
    test_create_transaction ()
```

Bases: auxi.modelling.financial.reporting.Report

Report a balance sheet of a general ledger.

Parameters

- data_source The object to report on.
- **end** The end date to generate the report for.
- output_path The path to write the report file to.

Bases: auxi.modelling.financial.reporting.Report

Report on a general ledger structure.

Parameters

- data_source The object to report on.
- output_path The path to write the report file to.

end=datetime.date(9999,

12, 31), out-

put_path=None)

Bases: auxi.modelling.financial.reporting.Report

Report an income statement of a general ledger.

Parameters

- data_source The object to report on.
- **end** The start date to generate the report for.
- **end** The end date to generate the report for.
- **component_path** The path of the component to filter the report's transactions by.
- output_path The path to write the report file to.

Bases: auxi.core.reporting.Report

Base class for reports.

Parameters

- data_source The object to report on.
- output_path The path to write the report file to.

```
class auxi.modelling.financial.reporting. ReportType
     Bases: enum.Enum
```

Represents a report type, e.g. balance sheet or income statement.

```
balance_sheet = <ReportType.balance_sheet: (1,)>
cash_flow = <ReportType.cash_flow: 4>
income_statement = <ReportType.income_statement: (2,)>
transaction_list = <ReportType.transaction_list: (3,)>
```

```
class auxi.modelling.financial.reporting. TransactionList (data_source,
                                                                         start = datetime.date(1,
                                                                         1,
                                                                         1),
                                                                         end=datetime.date(9999,
                                                                         12,
                                                                         31),
                                                                         com-
                                                                         po-
                                                                         nent_path='',
                                                                         out-
                                                                         put_path=None)
```

Bases: auxi.modelling.financial.reporting.Report

Report on a list of transactions.

Parameters

- data_source The object to report on.
- **end** The start date to generate the report for.
- **end** The end date to generate the report for.
- component_path The path of the component to filter the report's transactions by.
- **output_path** The path to write the report file to.

auxi.modelling.financial.reporting_test module

This module provides testing code for the auxi.modelling.financial.des module.

```
class auxi.modelling.financial.reporting_test. GeneralLedgerStructureUnitTes
     Bases: unittest.case.TestCase
    Tester for the auxi.modelling.financial.reporting.GeneralLedgerStructure class.
     setUp ( )
    test__generate_table_ ( )
class auxi.modelling.financial.reporting_test. TransactionListUnitTester ( me
     Bases: unittest.case.TestCase
    Tester for the auxi.modelling.financial.reporting.TransactionList class.
     setUp ( )
```

Module contents

auxi.modelling.financial Package

test__generate_table_ ()

The purpose of this package is to provide common classes and functions for the the auxi.modelling.financial namespace

auxi.modelling.process package

Subpackages

auxi.modelling.process.materials package

Submodules

auxi.modelling.process.materials.chem module

This module provides classes to work with materials and material packages that are described with chemical compositions.

Bases: auxi.core.objects.NamedObject

A material consisting of multiple chemical compounds.

Parameters

- name The material's name.
- **file_path** The path of the material definition file.
- **description** the material's description

The format of the text file is as follows:

- •The lines are space separated. The values in a line are separated by one or more spaces.
- •The first line is a heading line.
- •All subsequent lines contain a compound formula, followed by mass fractions.
- •The first column lists the compounds in the material.
- •All subsequent columns describe assays of the material.

The following is an example of a material text file:

Compound	IlmeniteA	IlmeniteB	IlmeniteC
A1203	0.01160	0.01550	0.00941
CaO	0.00022	0.00001	0.00017
Cr203	0.00008	0.00022	0.00011
Fe203	0.20200	0.47300	0.49674
Fe304	0.00000	0.00000	0.00000

FeO	0.27900	0.19100	0.00000	
K20	0.00004	0.00001	0.00005	
MgO	0.01040	0.00580	0.01090	
MnO	0.00540	0.00480	0.00525	
Na20	0.00007	0.00005	0.00031	
P4010	0.00001	0.00032	0.00015	
SiO2	0.00850	0.00490	0.01744	
TiO2	0.47700	0.29400	0.45949	
V205	0.00360	0.00800	0.00000	

add_assay (name, assay)

Add an assay to the material.

Parameters

- name The name of the new assay.
- **assay** A list containing the compound mass fractions for the assay. The sequence of the assay's elements must correspond to the sequence of the material's compounds.

create_empty_assay ()

Create an empty array to store an assay. The array's length will be equal to the number of compounds in the material.

Returns A floating point array.

create_package (assay=None, mass=0.0, normalise=True)

Create a MaterialPackage based on the specified parameters.

Parameters

- **assay** The name of the assay based on which the package must be created.
- mass [kg] The mass of the package.
- **normalise** Indicates whether the assay must be normalised before creating the package.

Returns The created MaterialPackage.

get_assay_total (name)

Calculate the total of the specified assay.

Parameters name – The name of the assay.

Returns The total mass fraction of the specified assay.

get_compound_index (compound)

Determine the index of the specified compound.

Parameters compound – The formula and phase of the specified compound, e.g. 'Fe2O3[S1]'.

Returns The index of the specified compound.

Bases: auxi.core.objects.Object

A package of a material consisting of multiple chemical compounds.

Parameters

- material A reference to the Material to which self belongs.
- **compound_masses** [kg] The masses of the compounds in the package.

```
add_to ( other)
```

Add another chem material package to this material package.

Parameters other – The other material package.

clear()

Set all the compound masses in the package to zero.

clone ()

Create a complete copy of self.

Returns A Material Package that is identical to self.

```
extract ( other)
```

Extract 'other' from self, modifying self and returning the extracted material as a new package.

Parameters other – Can be one of the following:

- float: A mass equal to other is extracted from self. Self is reduced by other and the extracted package is returned as a new package.
- tuple (compound, mass): The other tuple specifies the mass of a compound to be extracted. It is extracted from self and the extracted mass is returned as a new package.
- string: The 'other' string specifies the compound to be extracted. All of the mass of that compound will be removed from self and a new package created with it.

Returns A new material package containing the material that was extracted from self.

```
get_assay ( )
```

Determine the assay of self.

Returns [mass fractions] An array containing the assay of self.

```
get_compound_mass ( compound)
```

Get the mass of the specified compound in the package.

Parameters compound – The formula of the compound, e.g. Fe2O3.

Returns [kg]

get_compound_mass_fraction (compound)

Get the mass fraction of the specified compound in self.

Parameters compound – The formula and phase of the compound, e.g. Fe2O3.

Returns []

```
get_element_mass ( element)
```

Determine the masses of elements in the package.

Returns [kg] An array of element masses. The sequence of the elements in the result corresponds with the sequence of elements in the element list of the material.

```
get_element_mass_dictionary ( )
```

Determine the masses of elements in the package and return as a dictionary.

Returns [kg] A dictionary of element symbols and masses.

```
get_element_masses ()
```

Get the masses of elements in the package.

Returns [kg] An array of element masses. The sequence of the elements in the result corresponds with the sequence of elements in the element list of the material.

```
get_mass()
```

Get the mass of the package.

Returns [kg]

auxi.modelling.process.materials.chem test module

This module provides testing code for classes in the chem module.

Tester for the auxi.modelling.process.materials.chemistry.material.Material class.

```
add_incompatible_packages ( )
setUp ( )
test_add_operator_1 ( )
   other = MaterialPackage
```

Test whether the add operator calculates the resulting package correctly. Results were checked against FactSage results. They are not exactly the same, since the magnetic and other non-cp contributions are omitted by the thermo module.

```
test_add_operator_2 ()
```

Tests the scenario when a 'mix' is created and two packages 'mixed' into the mix package.

```
the thermo module.
    test_clone ()
    test_constructor ( )
    test extract 1 ()
    test_get_assay ( )
    test_get_compound_mass ()
    test_get_element_masses ()
    test_get_mass()
    test_multiply_operator()
    test_subtract_operator_2 ( )
    test_subtract_operator_3 ()
class auxi.modelling.process.materials.chem_test. ChemMaterialUnitTester ( me
    Bases: unittest.case.TestCase
    Tester for the auxi.modelling.process.materials.chem.Material class.
    assertAlmostEqual (first, second, places=14, msg=None, delta=None)
    setUp ( )
    test_add_assay ( )
    test_constructor ( )
    test_create_empty_assay ( )
    test_create_package ( )
    test_get_assay_total ()
    test_get_compound_index ( )
```

other = tuple (compound, mass) Test whether the add operator calculates the resulting package correctly. Results were checked against FactSage results. They are not exactly the same, since the magnetic and other non-cp contributions are omitted by

This module provides a material class that can do thermochemical calculations.

Bases: auxi.core.objects.NamedObject

auxi.modelling.process.materials.core module

test_add_operator_3 ()

Represents a material consisting of multiple chemical compounds, having the ability to do thermochemical calculations.

Parameters

- name A name for the material.
- **state_of_matter** The material's state of matter, e.g. liquid.
- **file_path** The location of the file containing the material's data.
- **description** the material's description

```
Cp ( **state)
```

Calculate the Cp value given the material state.

Parameters **state – material state

Returns float

alpha (**state)

Calculate the alpha value given the material state.

Parameters **state – material state

Returns float

beta (**state)

Calculate the alpha value given the material state.

Parameters **state - material state

Returns float

k (**state)

Calculate the k value given the material state.

Parameters **state – material state

Returns float

mstate = None

State of matter.

mu (**state)

Calculate the mu value given the material state.

Parameters **state - material state

Returns float

nu (**state)

Calculate the nu value given the material state.

Parameters **state - material state

Returns float

rho (**state)

Calculate the rho value given the material state.

Parameters **state - material state

Returns float

auxi.modelling.process.materials.core test module

This module provides testing code for classes in the chem module.

```
{\bf class} \ {\tt auxi.modelling.process.materials.core\_test.} \ {\bf MaterialTester} \ ( \ {\it methodName=1} \\ {\bf Bases:} \ {\tt unittest.case.TestCase}
```

Tester for the auxi.modelling.process.materials.core.Material class.

```
test_beta ( )
test constructor ( )
```

auxi.modelling.process.materials.psd module

This module provides psd material and material package classes that can do size distribution calculations.

Dases. aux1.core.objects.Namedobject

Represents a particulate material consisting of multiple particle size classes.

Parameters

- name The material's name.
- **file_path** The path of the material definition file.
- **description** the material's description

The format of the text file is as follows:

- •The lines are space separated. The values in a line are separated by one or more spaces.
- •The first line is a heading line.
- •All subsequent lines contain a particle size, followed by mass fractions.
- •Particle sizes are indicated in [meter].
- •The first column lists the particle sizes in the material. Each class must be interpreted as "mass fraction retained". In other words if the size class is indicated as 307.2E-3, it means that it is the class of material retained on a 307.2mm screen, and can also be though of as +307.2mm material. The first size class represents the largest particles. The final size class should be zero, as it represents all material that passed through the smallest aperture screen.

•All subsequent columns describe assays of the material.

The following is an example of a material text file:

```
Compound
          FeedA MillCharge
307.2E-3
         0.20
                0.02
108.6E-3
          0.18
                0.06
38.4E-3 0.17
                0.04
13.6E-3
         0.07 0.03
4.8E-3
          0.13
                0.03
1.7E-3
          0.07 0.04
600.0E-6
          0.06 0.18
210.0E-6 0.02 0.50
75.0E-6
          0.10
                0.10
0.0E0
          0.00
                0.00
```

add_assay (name, assay)

Add an assay to the material.

Parameters

- name The name of the new assay.
- **assay** A numpy array containing the size class mass fractions for the assay. The sequence of the assay's elements must correspond to the sequence of the material's size classes.

create_empty_assay ()

Create an empty array to store an assay. The array's length will be equal to the number of size classes in the material.

Returns A floating point array.

```
create_package ( assay=None, mass=0.0, normalise=True)
```

Create a MaterialPackage based on the specified parameters.

Parameters

- **assay** The name of the assay based on which the package must be created.
- mass [kg] The mass of the package.
- **normalise** Indicates whether the assay must be normalised before creating the package.

Returns The created MaterialPackage.

```
get_assay_total ( name)
```

Calculate the total of the specified assay.

Parameters name – The name of the assay.

Returns The total mass fraction of the specified assay.

```
get_size_class_index ( size_class)
```

Determine the index of the specified size class.

Parameters size_class – The formula and phase of the specified size class, e.g. 'Fe2O3[S1]'.

Returns The index of the specified size class.

Bases: auxi.core.objects.Object

A package of a material consisting of multiple particle size classes.

Properties defined here:

Parameters

- material A reference to the Material to which self belongs.
- **size_class_masses** [kg] [kg] The masses of the size classes in the package.

```
add_to ( other)
```

Add another psd material package to this material package.

Parameters other – The other material package.

```
clear ()
```

Set all the size class masses in the package to zero.

```
clone ( )
```

Create a complete copy of self.

Returns A Material Package that is identical to self.

```
extract ( other)
```

Extract 'other' from self, modifying self and returning the extracted material as a new package.

Parameters other – Can be one of the following:

- float: A mass equal to other is extracted from self. Self is reduced by other and the extracted package is returned as a new package.
- tuple (size class, mass): The other tuple specifies the mass of a size class to be extracted. It is extracted from self and the extracted mass is returned as a new package.
- string: The 'other' string specifies the size class to be extracted. All of the mass of that size class will be removed from self and a new package created with it.

Returns A new material package containing the material that was extracted from self.

```
get assay()
```

Determine the assay of self.

Returns [mass fractions] An array containing the assay of self.

```
get_mass()
          Determine the mass of self.
          returns: [kg] The mass of self.
     get_size_class_mass ( size_class)
          Determine the mass of the specified size class in self.
              Parameters size_class - The formula and phase of the size class,
                  e.g. 'Fe2O3[S1]'
              Returns [kg] The mass of the size class in self.
     get_size_class_mass_fraction ( size_class)
          Determine the mass fraction of the specified size class in self.
              Parameters size_class - The formula and phase of the size class,
                  e.g. 'Fe2O3[S1]'
              Returns The mass fraction of the size class in self.
auxi.modelling.process.materials.psd_test module
This module provides testing code for the psdmaterial module.
@author: Ex Mente Technologies (Pty) Ltd
class auxi.modelling.process.materials.psd_test. PsdMaterialPackageUnitTeste
     Bases: unittest.case.TestCase
     Tester for the auxi.modelling.process.materials.psd.material.MaterialPackage class.
     setUp ()
     test_add_operator_1 ( )
          other = MaterialPackage Test whether the add operator calculates the resulting
          package correctly.
     test_add_operator_3 ()
          other = tuple (size_class, mass) Test whether the add operator calculates the result-
          ing package correctly. Results were checked against FactSage results. They are not
          exactly the same, since the magnetic and other non-cp contributions are omitted by
          the thermo module.
     test clone ()
     test constructor ()
     test_extract_1 ( )
     test_get_assay ( )
     test_get_mass()
     test_get_size_class_mass()
```

test_multiply_operator()

```
test_subtract_operator_3 ()
class auxi.modelling.process.materials.psd_test. PsdMaterialUnitTester ( method
    Bases: unittest.case.TestCase
    Tester for the auxi.modelling.process.materials.psd.Material class.
    setUp ()
    test_add_assay ()
    test_constructor ()
    test_create_empty_assay ()
    test_create_package ()
    test_get_assay_total ()
```

test_subtract_operator_2 ()

auxi.modelling.process.materials.slurry module

test_get_size_class_index ()

This module provides material and material package classes that can do size distribution and slurry calculations.

Represents a particulate material consisting of multiple particle size classes.

Parameters

- name The material's name.
- **file_path** The path of the material definition file.
- **description** the material's description

The format of the text file is as follows:

- •The lines are space separated. The values in a line are separated by one or more spaces.
- •The first line is a heading line.
- •The second line contains the density of the solid material.
- •The third line contains the water fraction of the slurry (wet basis).
- •All subsequent lines contain a particle size, followed by mass fractions (dry basis).
- •Particle sizes are indicated in [meter].

- •The first column lists the particle sizes in the material. Each class must be interpreted as "mass fraction retained". In other words if the size class is indicated as 307.2E-3, it means that it is the class of material retained on a 307.2mm screen, and can also be though of as +307.2mm material. The first size class represents the largest particles. The final size class should be zero, as it represents all material that passed through the smallest aperture screen.
- •All subsequent columns describe assays of the material.

The following is an example of a material text file:

SizeClass →WetMillCharge	_	DryMillCharge	WetFeedA	_	
solid_density → 1.0.		3.00	3.00	3.00	
H2O → 1.00	0.00	0.00	0.80	0.60	u
307.2E−3 → 0.00	0.20	0.02	0.20	0.02	u
108.6E−3 → 0.00	0.18	0.06	0.18	0.06	
38.4E-3 → 0.00	0.17	0.04	0.17	0.04	u
13.6E-3 → 0.00	0.07	0.03	0.07	0.03	
4.8E-3 → 0.00	0.13	0.03	0.13	0.03	
1.7E-3 → 0.00	0.07	0.04	0.07	0.04	٦
600.0E-6 → 0.00	0.06	0.18	0.06	0.18	u
210.0E-6 → 0.00	0.02	0.50	0.02	0.50	u
75.0E-6 → 0.00	0.10	0.09	0.10	0.09	u
0.0E0 → 0.00	0.00	0.00	0.00	0.00	

add_assay (name, solid_density, H2O_fraction, assay)
Add an assay to the material.

Parameters

- name The name of the new assay.
- **assay** A numpy array containing the size class mass fractions for the assay. The sequence of the assay's elements must correspond to the sequence of the material's size classes.

create_empty_assay ()

Create an empty array to store an assay. The array's length will be equal to the number of size classes in the material.

Returns A floating point array.

create_package (assay=None, mass=0.0, normalise=True)

Create a MaterialPackage based on the specified parameters.

Parameters

- **assay** The name of the assay based on which the package must be created.
- mass [kg] The mass of the package.
- **normalise** Indicates whether the assay must be normalised before creating the package.

Returns The created MaterialPackage.

```
get_assay_total ( name)
```

Calculate the total of the specified assay.

Parameters name – The name of the assay.

Returns The total mass fraction of the specified assay.

```
get_size_class_index ( size_class)
```

Determine the index of the specified size class.

Parameters size_class – The formula and phase of the specified size class, e.g. 'Fe2O3[S1]'.

Returns The index of the specified size class.

Bases: auxi.core.objects.Object

A package of a slurry material consisting of multiple particle size classes.

Parameters

- material A reference to the Material to which self belongs.
- **size_class_masses** [kg] [kg] The masses of the size classes in the package.

clear ()

Set all the size class masses and H20_mass in the package to zero and the solid_density to 1.0

clone ()

Create a complete copy of self.

Returns A Material Package that is identical to self.

```
extract ( other)
```

Extract 'other' from self, modifying self and returning the extracted material as a new package.

Parameters other – Can be one of the following:

- float: A mass equal to other is extracted from self. Self is reduced by other and the extracted package is returned as a new package.
- tuple (size class, mass): The other tuple specifies the mass of a size class to be extracted. It is extracted from self and the extracted mass is returned as a new package.
- string: The 'other' string specifies the size class to be extracted. All of the mass of that size class will be removed from self and a new package created with it.

Returns A new material package containing the material that was extracted from self.

```
get_assay ( )
```

Determine the assay of self.

Returns [mass fractions] An array containing the assay of self.

```
get_density()
```

Determine the density of self.

```
get_mass()
```

Determine the mass of self.

Returns [kg] The mass of self.

```
get_mass_fraction_solids ( )
```

Determine the mass fraction of the solids of self.

```
get_size_class_mass ( size_class)
```

Determine the mass of the specified size class in self.

Parameters size_class – The formula and phase of the size class, e.g. 'Fe2O3[S1]'

Returns [kg] The mass of the size class in self.

```
get_size_class_mass_fraction ( size_class)
```

Determine the mass fraction of the specified size class in self.

Parameters size_class – The formula and phase of the size class, e.g. Fe2O3[S1]

Returns The mass fraction of the size class in self.

```
get_solid_mass()
```

Determine the solid mass of self.

Returns [kg] The solid mass of self.

```
get_volume ( )
```

Determine the volume of self.

```
get_volume_fraction_solids ( )
```

Determine the volume fraction of the solids of self.

auxi.modelling.process.materials.slurry test module

```
This module provides testing code for the psdslurrymaterial module.
```

```
@author: Ex Mente Technologies (Pty) Ltd
class auxi.modelling.process.materials.slurry_test. SlurryMaterialPackageUni
     Bases: unittest.case.TestCase
     setUp ()
    test___str__ ( )
    test_add_operator_1 ( )
         other = MaterialPackage Test whether the add operator calculates the resulting
         package correctly.
    test_add_operator_3 ()
         other = tuple (size_class, mass) Test whether the add operator calculates the result-
         ing package correctly. Results were checked against FactSage results. They are not
         exactly the same, since the magnetic and other non-cp contributions are omitted by
         the thermo module.
    test clone ()
    test constructor ()
    test extract 1 ()
    test_get_assay ( )
    test_get_mass()
    test_get_size_class_mass()
    test_multiply_operator()
    test_subtract_operator_2 ( )
    test_subtract_operator_3 ()
class auxi.modelling.process.materials.slurry test. SlurryMaterialUnitTester
     Bases: unittest.case.TestCase
    Tester for the auxi.modelling.process.materials.slurry.Material class.
     setUp ()
    test_add_assay ( )
    test_constructor ( )
    test_create_empty_assay ( )
    test_create_package ( )
    test_get_assay_total ()
```

test_get_size_class_index()

auxi.modelling.process.materials.thermo module

This module provides a material class that can do thermochemical calculations.

Bases: auxi.core.objects.NamedObject

Represents a material consisting of multiple chemical compounds, having the ability to do thermochemical calculations.

Parameters

- name A name for the material.
- **file_path** The location of the file containing the material's data.
- **description** the material's description

The format of the text file is as follows:

- •The items in a line are separated by one or more spaces or tabs.
- •The first line is a heading line. It contains the word "Compound" followed by zero or more assay names.
- •Subsequent lines contain a compound formula and phase, followed by a mass fraction for each assay.
- •The list of compounds and mass fractions can be ended off with a "#" character. This indicates that custom material properties follow below in the lines below the hash.
- •If a custom material property is defined, a value must be provided for each assay name. A price custom property is used as an example below.

The following is an example of a material text file:

Compound	IlmeniteA	IlmeniteB	IlmeniteC
Al203[S1]	0.01160	0.01550	0.00941
CaO[S]	0.00022	0.00001	0.00017
Cr203[S]	0.00008	0.00022	0.00011
Fe203[S1]	0.20200	0.47300	0.49674
Fe304[S1]	0.00000	0.00000	0.0000
Fe0[S1]	0.27900	0.19100	0.0000
K20[S]	0.00004	0.00001	0.00005
Mg0[S]	0.01040	0.00580	0.01090
MnO[S]	0.00540	0.00480	0.00525
Na20[S1]	0.00007	0.00005	0.00031
P4010[S]	0.00001	0.00032	0.00015
SiO2[S1]	0.00850	0.00490	0.01744
TiO2[S1]	0.47700	0.29400	0.45949

V205[S]	0.00360	0.00800	0.00000
#			
Price[USD	/kg] 1.2	1.3	1.1

add_assay (name, assay)

Add an assay to the material.

Parameters

- name Assay name.
- **assay** Numpy array containing the compound mass fractions for the assay. The sequence of the assay's elements must correspond to the sequence of the material's compounds.

compound_count = None

The number of chemical compounds in the material.

compounds = None

The material's list of chemical compounds.

converted_assays = None

A dictionary containing converted assays for this material.

create_empty_assay ()

Create an empty array to store an assay.

The array's length will be equal to the number of compounds in the material.

Returns Empty assay array.

create_package (assay=None, mass=0.0, P=1.0, T=25.0, normalise=True) Create a MaterialPackage based on the specified parameters.

Parameters

- **assay** Name of the assay to be used to create the package.
- mass Package mass. [kg]
- P Package pressure. [atm]
- **T** Package temperature. [°C]
- **normalise** Indicates whether the assay must be normalised before creating the package.

Returns MaterialPackage object.

description = None

The material's description.

get_assay_total (name)

Calculate the total/sum of the specified assay's mass fractions.

Parameters name – Assay name.

Returns Total mass fraction.

```
get_compound_index ( compound)
```

Determine the specified compound's index.

Parameters compound – Formula and phase of a compound, e.g. "Fe2O3[S1]".

Returns Compound index.

name = None

The material's name.

raw_assays = None

A dictionary containing raw assays for this material.

```
class auxi.modelling.process.materials.thermo. MaterialPackage ( material, com-pound\_masses, P=1.0, T=25.0)
```

Bases: auxi.core.objects.Object

Represents a quantity of material consisting of multiple chemical compounds, having a specific mass, pressure, temperature and enthalpy.

Parameters

- material A reference to the Material to which self belongs.
- compound_masses Package compound masses. [kg]
- **P** Package pressure. [atm]
- **T** Package temperature. [°C]

Н

Get the enthalpy of the package.

Returns Enthalpy. [kWh]

P

Determine the pressure of the package.

Returns Pressure. [atm]

Т

Get the temperature of of the package.

Returns Temperature. [°C]

amount

Determine the sum of mole amounts of all the compounds.

Returns Amount. [kmol]

clear ()

Set all the compound masses in the package to zero. Set the pressure to 1, the temperature to 25 and the enthalpy to zero.

clone ()

Create a complete copy of the package.

Returns A new MaterialPackage object.

extract (other)

Extract 'other' from this package, modifying this package and returning the extracted material as a new package.

Parameters other – Can be one of the following:

- float: A mass equal to other is extracted from self. Self is reduced by other and the extracted package is returned as a new package.
- tuple (compound, mass): The other tuple specifies the mass of a compound to be extracted. It is extracted from self and the extracted mass is returned as a new package.
- string: The 'other' string specifies the compound to be extracted. All of the mass of that compound will be removed from self and a new package created with it.
- Material: The 'other' material specifies the list of compounds to extract.

Returns New MaterialPackage object.

get_assay ()

Determine the assay of the package.

Returns Array of mass fractions.

get_compound_amount (compound)

Determine the mole amount of the specified compound.

Returns Amount. [kmol]

get_compound_amounts ()

Determine the mole amounts of all the compounds.

Returns List of amounts. [kmol]

get_compound_mass (compound)

Determine the mass of the specified compound in the package.

Parameters compound – Formula and phase of a compound, e.g. "Fe2O3[S1]".

Returns Mass. [kg]

get_element_mass (element)

Determine the mass of the specified elements in the package.

Returns Masses. [kg]

get_element_mass_dictionary ()

Determine the masses of elements in the package and return as a dictionary.

Returns Dictionary of element symbols and masses. [kg]

```
get_element_masses ( elements=None)
```

Determine the masses of elements in the package.

Returns Array of element masses. [kg]

mass

Get the mass of the package.

Returns [kg]

auxi.modelling.process.materials.thermo_test module

This module provides testing code for classes in the thermo module.

Unit tester for the auxi.modelling.process.materials.thermo.MaterialPackage class.

```
add_incompatible_packages ()
setUp ()
test___str__ ()
test add operator 1 ()
```

Test whether the add operator calculates the resulting package correctly. Results were checked against FactSage results. They are not exactly the same, since the magnetic and other non-cp contributions are omitted by the thermo module.

```
test_add_operator_2 ( )
test_add_operator_3 ( )
  other = tuple (compound, mass)
```

Test whether the add operator calculates the resulting package correctly. Results were checked against FactSage results. They are not exactly the same, since the magnetic and other non-cp contributions are omitted by the thermo module.

```
test_add_operator_4 ( )
  other = tuple (compound, mass, temperature)
```

Test whether the add operator calculates the resulting package correctly. Results were checked against FactSage results. They are not exactly the same, since the magnetic and other non-cp contributions are omitted by the thermo module.

```
test_clone ()
test_constructor ()
test_extract_1 ()
test_extract_2 ()
test_extract_3 ()
```

```
test_get_T ()
    test_get_assay ( )
    test_get_compound_mass ( )
    test_get_element_masses ( )
    test_mass ()
    test_multiply_operator()
    test_set_H ( )
    test_set_T ( )
class auxi.modelling.process.materials.thermo_test. ThermoMaterialUnitTester
    Bases: unittest.case.TestCase
    Unit tester for the auxi.modelling.process.materials.thermo.Material class.
    setUp ( )
    test_add_assay ( )
    test constructor ()
    test_create_empty_assay ( )
    test_create_package ( )
    test_get_assay_total ()
```

Module contents

test_get_compound_index ()

test_get_H ()

test_get_P ()

This package provides classes and functions for working with different types of materials and material packages.

Module contents

The purpose of this package is to provide common classes and functions for the the auxi.modelling.process namespace

Module contents

The purpose of this package is to provide common classes and functions for the auxi.modelling namespace

auxi.tools package

Subpackages

auxi.tools.chemistry package

Submodules

auxi.tools.chemistry.stoichiometry module

This module provides a number of functions for doing stoichiometry calculations.

Bases: auxi.core.objects.Object

An element in the periodic table.

Parameters

- **period** Period to which the element belongs.
- **group** Group to which the element belongs.
- atomic_number Number of protons in the element's nucleus.
- **symbol** Element's symbol.
- molar_mass [kg/kmol] Element's standard atomic mass.

auxi.tools.chemistry.stoichiometry. amount (compound, mass)

Calculate the number of moles in the specified mass of a chemical compound.

Parameters

- **compound** Formula and phase of a compound, e.g. 'Fe2O3[S1]'. The phase may be omitted.
- mass [kg]

Returns Amount. [kmol]

auxi.tools.chemistry.stoichiometry.amount_fractions (masses) Calculate the mole fractions from the specified compound masses.

Parameters masses – [kg] dictionary, e.g. {'SiO2': 3.0, 'FeO', 1.5}

Returns [mole fractions] dictionary

auxi.tools.chemistry.stoichiometry. amounts (masses) Calculate the amounts from the specified compound masses.

Parameters masses – [kg] dictionary, e.g. {'SiO2': 3.0, 'FeO', 1.5}

Returns [kmol] dictionary

```
auxi.tools.chemistry.stoichiometry.convert_compound ( mass, source, target, ele-
```

Convert the specified mass of the source compound to the target using element as basis.

Parameters

- mass Mass of from_compound. [kg]
- **source** Formula and phase of the original compound, e.g. 'Fe2O3[S1]'.
- target Formula and phase of the target compound, e.g. 'Fe[S1]'.
- **element** Element to use as basis for the conversion, e.g. 'Fe' or 'O'.

Returns Mass of target. [kg]

```
auxi.tools.chemistry.stoichiometry.element_mass_fraction ( compound, el- e- ment)
```

Determine the mass fraction of an element in a chemical compound.

Parameters

- **compound** Formula of the chemical compound, 'FeCr2O4'.
- element Element, e.g. 'Cr'.

Returns Element mass fraction.

Determine the mass fractions of a list of elements in a chemical compound.

Parameters

- **compound** Formula and phase of a chemical compound, e.g. 'Fe2O3[S1]'.
- elements List of elements, ['Si', 'O', 'Fe'].

Returns Mass fractions.

```
auxi.tools.chemistry.stoichiometry.elements (compounds)

Determine the set of elements present in a list of chemical compounds.
```

The list of elements is sorted alphabetically.

Parameters compounds – List of compound formulas and phases, e.g. ['Fe2O3[S1]', 'Al2O3[S1]'].

Returns List of elements.

auxi.tools.chemistry.stoichiometry. mass (compound, amount) Calculate the mass of the specified amount of a chemical compound.

Parameters

- **compound** Formula and phase of a compound, e.g. 'Fe2O3[S1]'. The phase may be omitted.
- amount [kmol]

Returns Mass. [kg]

auxi.tools.chemistry.stoichiometry.mass_fractions (amounts)
Calculate the mole fractions from the specified compound masses.

Parameters masses – [kg] dictionary, e.g. {'SiO2': 3.0, 'FeO', 1.5}

Returns [mass fractions] dictionary

auxi.tools.chemistry.stoichiometry.masses (*amounts*) Calculate the masses from the specified compound amounts.

Parameters masses – [kmol] dictionary, e.g. {'SiO2': 3.0, 'FeO', 1.5}

Returns [kg] dictionary

auxi.tools.chemistry.stoichiometry.molar_mass (compound="') Determine the molar mass of a chemical compound.

The molar mass is usually the mass of one mole of the substance, but here it is the mass of 1000 moles, since the mass unit used in pmpy is kg.

Parameters compound – Formula of a chemical compound, e.g. 'Fe2O3'.

Returns Molar mass. [kg/kmol]

Determine the stoichiometry coefficient of an element in a chemical compound.

Parameters

- compound Formula of a chemical compound, e.g. 'SiO2'.
- element Element, e.g. 'Si'.

Returns Stoichiometry coefficient.

```
auxi.tools.chemistry.stoichiometry.stoichiometry_coefficients(compound, el-
e-
ments)
```

Determine the stoichiometry coefficients of the specified elements in the specified chemical compound.

Parameters

- compound Formula of a chemical compound, e.g. 'SiO2'.
- elements List of elements, e.g. ['Si', 'O', 'C'].

Returns List of stoichiometry coefficients.

auxi.tools.chemistry.stoichiometry test module

This module contains all the code used to test the testee module.

```
class auxi.tools.chemistry.stoichiometry_test. StoichFunctionTester ( methodNa Bases: unittest.case.TestCase
```

The function tester for the stoichiometry module.

```
\verb|assertAlmostEqual| (first, second, places=14, msg=None, delta=None)|
```

```
test_amount ( )
```

Test whether the amount of a compound is calculated correctly.

```
test_amount_fractions ()
```

Test whether compound amount fractions are calculated correctly.

```
test amounts ()
```

Test whether compound amounts are calculated correctly.

```
test_convert_compound ( )
```

Test whether compound conversions are calculated correctly.

```
test_element_mass_fraction ( )
```

Test whether an element mass fraction is calculated correctly.

```
test_element_mass_fractions()
```

Test whether a list of element mass fraction is calculated correctly.

```
test_elements()
```

Test whether the set of elements in a list of compounds is calculated correctly.

```
test_invalid_characters()
```

Test whether an exception is raised when a compound formula contains an invalid character.

```
test_mass ()
```

Test whether the mass of a compound is calculated correctly.

```
test_mass_fractions()
```

Test whether compound mass fractions are calculated correctly.

```
test_masses ( )
```

Test whether compound masses are calculated correctly.

```
test_molar_mass()
```

Test whether the molar mass of a compound is calculated correctly.

test_stoichiometry_coefficient()

Test whether the stoichiometry coefficient of a specified element in a compound is calculated correctly.

test_stoichiometry_coefficients ()

Test whether the stoichiometry coefficients of a specified list of elements in a compound is calculated correctly.

auxi.tools.chemistry.thermochemistry module

This module provides classes and functions for doing thermochemical calculations.

```
class auxi.tools.chemistry.thermochemistry.Compound (dictionary)
    Bases: auxi.core.objects.Object
```

Represents a chemical compound.

Parameters dictionary – Dictionary containing the data required to initialise the compound.

Cp (*phase*, *temperature*)

Calculate the heat capacity of a phase of the compound at a specified temperature.

Parameters

- phase A phase of the compound, e.g. 'S', 'L', 'G'.
- temperature [K]

Returns [J/mol/K] Heat capacity.

G (*phase*, *temperature*)

Calculate the Gibbs free energy of a phase of the compound at a specified temperature.

Parameters

- phase A phase of the compound, e.g. 'S', 'L', 'G'.
- temperature [K]

Returns [J/mol] Gibbs free energy.

H (phase, temperature)

Calculate the enthalpy of a phase of the compound at a specified temperature.

Parameters

- phase A phase of the compound, e.g. 'S', 'L', 'G'.
- temperature [K]

Returns [J/mol] Enthalpy.

S (*phase*, *temperature*)

Calculate the enthalpy of a phase of the compound at a specified temperature.

Parameters

- phase A phase of the compound, e.g. 'S', 'L', 'G'.
- temperature [K]

Returns [J/mol/K] Entropy.

formula = None

Chemical formula, e.g. 'Fe', 'CO2'.

get_phase_list()

Get a list of the compound's phases.

Returns List of phases.

```
get_reference ( )
```

molar mass = None

Molar mass. [kg/mol]

auxi.tools.chemistry.thermochemistry.**Cp** (compound_string, temperature, mass=1.0)

Calculate the heat capacity of the compound for the specified temperature and mass.

Parameters

- **compound_string** Formula and phase of chemical compound, e.g. 'Fe2O3[S1]'.
- temperature [°C]
- mass [kg]

Returns [kWh/K] Heat capacity.

```
class auxi.tools.chemistry.thermochemistry.CpRecord (dictionary)
    Bases: auxi.core.objects.Object
```

A heat capacity (Cp) equation record for a compound phase over a specific temperature range.

Parameters dictionary – A dictionary containing the data required to initialise the phase.

Cp (temperature)

Calculate the heat capacity of the compound phase.

Parameters temperature – [K]

Returns [J/mol/K] Heat capacity.

H (temperature)

Calculate the portion of enthalpy of the compound phase covered by this Cp record.

Parameters temperature - [K]

Returns [J/mol] Enthalpy.

S (temperature)

Calculate the portion of entropy of the compound phase covered by this Cp record.

Parameters temperature – [K]

Returns Entropy. [J/mol/K]

Tmax = None

[K] The maximum temperature of the range covered by this record.

Tmin = None

[K] The minimum temperature of the range covered by this record.

auxi.tools.chemistry.thermochemistry. ${\bf G}$ ($compound_string$, temperature, mass=1.0)

Calculate the Gibbs free energy of the compound for the specified temperature and mass.

Parameters

- **compound_string** Formula and phase of chemical compound, e.g. 'Fe2O3[S1]'.
- temperature [°C]
- mass [kg]

Returns [kWh] Gibbs free energy.

auxi.tools.chemistry.thermochemistry.**H** (*compound_string*, *temperature*, *mass=1.0*)

Calculate the enthalpy of the compound for the specified temperature and mass.

Parameters

- **compound_string** Formula and phase of chemical compound, e.g. 'Fe2O3[S1]'.
- temperature [°C]
- mass-[kg]

Returns [kWh] Enthalpy.

class auxi.tools.chemistry.thermochemistry.Phase (dictionary)
 Bases: auxi.core.objects.NamedObject

A phase of a chemical compound.

Parameters dictionary – Dictionary containing the data required to initialise the phase.

Cp (temperature)

Calculate the heat capacity of the compound phase at the specified temperature.

Parameters temperature – [K]

Returns [J/mol/K] The heat capacity of the compound phase.

DHref = None

[J/mol] The formation enthalpy of the phase at Tref.

G (temperature)

Calculate the heat capacity of the compound phase at the specified temperature.

Parameters temperature - [K]

Returns [J/mol] The Gibbs free energy of the compound phase.

H (temperature)

Calculate the enthalpy of the compound phase at the specified temperature.

Parameters temperature – [K]

Returns [J/mol] The enthalpy of the compound phase.

S (temperature)

Calculate the entropy of the compound phase at the specified temperature.

Parameters temperature – [K]

Returns [J/mol/K] The entropy of the compound phase.

Sref = None

[J/mol/K] The standard entropy of the phase at Tref.

Tref = None

[K] The reference temperature of the phase.

name = None

The phase's name, e.g. solid, liquid, gas, etc.

symbol = None

The phase's symbol, e.g. S1 = solid 1, L = liquid, etc.

auxi.tools.chemistry.thermochemistry.**S** (compound_string, temperature, mass=1.0)

Calculate the entropy of the compound for the specified temperature and mass.

Parameters

- **compound_string** Formula and phase of chemical compound, e.g. 'Fe2O3[S1]'.
- temperature [°C]
- mass [kg]

Returns [kWh/K] Entropy.

- auxi.tools.chemistry.thermochemistry.get_datafile_references ()
 Retrieve all the references used by the datafiles.
- auxi.tools.chemistry.thermochemistry.list_compounds ()
 List all compounds that are currently loaded in the thermo module, and their phases.
- auxi.tools.chemistry.thermochemistry.load_data_auxi (path='') Load all the thermochemical data auxi files located at a path.

Parameters path – Path at which the data files are located.

auxi.tools.chemistry.thermochemistry.load_data_factsage (path='') Load all the thermochemical data factsage files located at a path.

Parameters path – Path at which the data files are located.

```
auxi.tools.chemistry.thermochemistry.molar_mass ( compound) Determine the molar mass of a chemical compound.
```

Parameters compound – Formula of a chemical compound, e.g. 'Fe2O3'.

Returns [kg/mol] Molar mass.

Writes a compound to an auxi file at the specified directory.

Parameters

- **dir** The directory.
- **compound** The compound.

auxi.tools.chemistry.thermochemistry_test module

This module provides testing code for the thermochemistry module.

```
test_cp()
test_G()
test_H()
test_S()
test_compound_get_phase_list()
test_cpRecord___str__()
test_get_datafile_references()
test_get_reference()
test_load_data_auxi()
test_molar_mass()
test_phase___str__()
```

Module contents

This package provides chemistry-related tools to make various tasks quick, easy and accurate.

auxi.tools.materialphysicalproperties package

Submodules

auxi.tools.materialphysicalproperties.core module

This module provides tools for calculating material physical properties.

```
class auxi.tools.materialphysicalproperties.core. DataSet (csvfilepath)
    Bases: object
```

Contains a data set used to create and/or test a material physical property model.

Parameters csvfilepath – path to the csv file that contains the data

```
create_template ( material, path, show=False)
```

Create a template csv file for a data set.

Parameters

- material the name of the material
- path the path of the directory where the file must be written
- **show** a boolean indicating whether the created file should be displayed after creation

```
class auxi.tools.materialphysicalproperties.core. Model (material,
                                                                       pro-
                                                                      prty,
                                                                       sym-
                                                                       bol,
                                                                       dis-
                                                                      play_symbol,
                                                                       units.
                                                                       state_schema,
                                                                       refer-
                                                                       ences,
                                                                       datasets)
```

Bases: auxi.core.objects.Object

Base class of models that describe the variation of a specific material physical property.

```
param material the name of the material being described, e.g.
```

param proprty the name of the property being described, e.g. "density"

param symbol the symbol of the property being described, e.g. "rho"

```
param display symbol the display symbol of the property being de-
                 scribed, e.g. "
     ho"
              param units the units used to express the property, e.g. "kg/m3"
              param references a list of literature references on which this model im-
                 plementation is based, e.g. ['lienhard2015', 'smith2006']
              param datasets a list of data sets on which this model implementation is
                 based, e.g. ['dataset-air-lienhard2015.csv']
     calculate ( **state)
          Base calculate method for models. Validates the material state parameter(s).
              Parameters **state – The material state
class auxi.tools.materialphysicalproperties.core. StateOfMatter
     Bases: enum. Enum
     gas = <StateOfMatter.gas: 4>
     gas_plasma = <StateOfMatter.gas_plasma: 12>
     liquid = <StateOfMatter.liquid: 2>
     liquid_gas_plasma = <StateOfMatter.liquid_gas_plasma: 14>
     liquid_plasma = <StateOfMatter.liquid_plasma: 10>
     plasma = <StateOfMatter.plasma: 8>
     solid = <StateOfMatter.solid: 1>
     solid_gas = <StateOfMatter.solid_gas: 5>
     solid_liquid = <StateOfMatter.solid_liquid: 3>
     solid_liquid_gas = <StateOfMatter.solid_liquid_gas: 7>
     solid_liquid_gas_plasma = <StateOfMatter.solid_liquid_gas_plasma: 15>
     solid_liquid_plasma = <StateOfMatter.solid_liquid_plasma: 11>
     solid_plasma = <StateOfMatter.solid_plasma: 9>
     unknown = <StateOfMatter.unknown: 0>
```

auxi.tools.materialphysicalproperties.core test module

This module contains all the code used to test the testee module.

```
class auxi.tools.materialphysicalproperties.core_test. DataSetTester ( methodN
    Bases: unittest.case.TestCase
```

The tester for the DataSet class.

```
test_construct ( )
    Test whether a data set is constructed successfully from a file.
test_create_template()
test_description ()
    Test whether the data set description is successfully loaded.
test material ()
     Test whether the material name is successfully loaded.
test_name ()
     Test whether the data set name is successfully loaded.
test parameter names ()
     Test whether the data set parameter names are successfully loaded.
test_parameter_symbols ()
    Test whether the data set parameter symbols are successfully loaded.
test_parameter_units ()
    Test whether the data set parameter units are successfully loaded.
test reference ()
```

auxi.tools.materialphysicalproperties.gases module

This module provides physical property data sets and models for gases.

Test whether the data set reference is successfully loaded.

auxi.tools.materialphysicalproperties.idealgas module

This module provides tools for calculating physical properties of an ideal gas.

A model that describes the variation in the thermal expansion coefficient of an ideal gas as a function of temperature.

```
calculate ( **state)
```

Calculate the thermal expansion coefficient at the specified temperature:

```
Parameters T − [K] temperature
```

Returns [1/K] thermal expansion coefficient

The **state parameter contains the keyword argument(s) specified above that are used to describe the state of the material.

```
class auxi.tools.materialphysicalproperties.idealgas. RhoT ( molar\_mass, P)

Bases: auxi.tools.materialphysicalproperties.core.Model
```

A model that describes the variation in density of an ideal gas as a function of temperature.

Parameters

- molar_mass [g/mol] average molar mass of the gas
- P [Pa] pressure

P = None

[Pa] pressure

```
calculate ( **state)
```

Calculate the density at the specified temperature.

Parameters T - [K] temperature

Returns [kg/m3] density

The **state parameter contains the keyword argument(s) specified above that are used to describe the state of the material.

mm = None

[kg/mol] average molar mass of the gas

```
class auxi.tools.materialphysicalproperties.idealgas. RhoTP ( molar_mass)
    Bases: auxi.tools.materialphysicalproperties.core.Model
```

A model that describes the variation in density of an ideal gas as a function of temperature and pressure.

Parameters molar_mass – [g/mol] average molar mass of the gas

```
calculate ( **state)
```

Calculate the density at the specified temperature and pressure.

Parameters

- **T** [K] temperature
- P [Pa] pressure

Returns [kg/m3] density

The **state parameter contains the keyword argument(s) specified above that are used to describe the state of the material.

mm = None

[kg/mol] average molar mass of the gas

```
class auxi.tools.materialphysicalproperties.idealgas. RhoTPx
    Bases: auxi.tools.materialphysicalproperties.core.Model
```

A model that describes the variation in density of an ideal gas as a function of temperature, pressure, and molar composition.

```
calculate ( **state)
```

Calculate the density at the specified temperature, pressure, and composition.

Parameters

- **T** [K] temperature
- P [Pa] pressure
- \mathbf{x} [mole fraction] dictionary of compounds and mole fractions

Returns [kg/m3] density

The **state parameter contains the keyword argument(s) specified above that are used to describe the state of the material.

auxi.tools.materialphysicalproperties.idealgas_test module

This module contains all the code used to test the testee module.

The tester for the BetaT class.

```
test_calculate ( )
```

Test whether the property value is calculated successfully.

```
test_call ()
```

Test whether the property value is calculated successfully via the __call__ magic method.

```
test_construct ( )
```

Test whether a model is constructed successfully.

The tester for the RhoTP class.

```
test_calculate ( )
```

Test whether the property value is calculated successfully.

```
test_call ()
```

Test whether the property value is calculated successfully via the __call__ magic method.

```
test_create ( )
```

Test whether a model is created successfully.

The tester for the RhoTPx class.

```
test calculate ()
```

Test whether the property value is calculated successfully.

```
test_call ()
          Test whether the property value is calculated successfully via the __call__ magic
          method.
     test_create ( )
          Test whether a model is created successfully.
class auxi.tools.materialphysicalproperties.idealgas_test. RhoTTester (method
     Bases: unittest.case.TestCase
     The tester for the RhoT class.
     test calculate()
          Test whether the property value is calculated successfully.
     test call ()
          Test whether the property value is calculated successfully via the __call__ magic
          method.
     test create ()
```

auxi.tools.materialphysicalproperties.liquids module

Test whether a model is created successfully.

This module provides physical property data sets and models for liquids.

auxi.tools.materialphysicalproperties.polynomial module

This module provides tools for calculating material physical properties.

```
class auxi.tools.materialphysicalproperties.polynomial. PolynomialModelT ( ma
```

Bases: auxi.tools.materialphysicalproperties.core.Model

A model that uses polynomial fits on experimental data to describe the variation of a specific material physical property as a function of temperature.

param material the name of the material being described, e.g. "Air"

pri Syl bodis plo un ref eren da co eff. **param proprty** the name of the property being described, e.g. "density"

param symbol the symbol of the property being described, e.g. "rho"

param display symbol the display symbol of the property being described, e.g. "

ho"

param units the units used to express the property, e.g. "kg/m3"

param references a list of literature references on which this model implementation is based, e.g. ['lienhard2015', 'smith2006']

param datasets a list of data sets on which this model implementation is based, e.g. ['dataset-air-lienhard2015.csv']

param coeffs polynomial coefficients sorted from highest to lowest power

```
calculate ( **state)
```

Calculate the material physical property at the specified temperature in the units specified by the object's 'property_units' property.

Parameters T – [K] temperature

Returns physical property value

```
create ( dataset, symbol, degree)
```

Create a model object from the data set for the property specified by the supplied symbol, using the specified polynomial degree.

Parameters

- dataset a DataSet object
- **symbol** the symbol of the property to be described, e.g. 'rho'
- degree the polynomial degree to use

Returns a new PolynomialModelT object

```
plot ( dataset, path, show=False)
```

auxi.tools.materialphysicalproperties.polynomial_test module

This module contains all the code used to test the testee module.

The tester for the PolynomialModelT class.

test calculate ()

Test whether the property value is calculated successfully.

test_call ()

Test whether the property value is calculated successfully via the __call__ magic method.

test_construct ()

Test whether a model is constructed successfully.

test_create ()

Test whether a model is created successfully from a data set.

test_plot()

Test whether the plots are created.

test_read ()

Test whether a model is created successfully by loading it from a json file.

test write ()

Test whether a model is successfully written to a json file.

auxi.tools.materialphysicalproperties.slags module

This module provides physical property data sets and models for slags.

A model that describes the variation in the dynamic viscosity of liquid slag as a function of temperature and composition expressed in mole fraction.

```
calculate ( **state)
```

Calculate dynamic viscosity at the specified temperature and composition:

Parameters

- $\mathbf{T} [K]$ temperature
- \mathbf{x} [mole fraction] composition dictionary , e.g. {'SiO2': 0.25, 'CaO': 0.25, 'MgO': 0.25, 'FeO': 0.25}

Returns [Pa.s] dynamic viscosity

The **state parameter contains the keyword argument(s) specified above that are used to describe the state of the material.

A model that describes the variation in the dynamic viscosity of liquid slag as a function of temperature and composition expressed in mass fraction.

```
calculate ( **state)
```

Calculate dynamic viscosity at the specified temperature and composition:

Parameters

- $\mathbf{T} [K]$ temperature
- **y** [mass fraction] composition dictionary , e.g. {'SiO2': 0.25, 'CaO': 0.25, 'MgO': 0.25, 'FeO': 0.25}

Returns [Pa.s] dynamic viscosity

The **state parameter contains the keyword argument(s) specified above that are used to describe the state of the material.

A model that describes the variation in the dynamic viscosity of liquid slag as a function of temperature and composition expressed in mole fraction.

```
calculate ( **state)
```

Calculate dynamic viscosity at the specified temperature and composition:

Parameters

- $\mathbf{T} [K]$ temperature
- **x** [mole fraction] composition dictionary, e.g. {'SiO2': 0.25, 'CaO': 0.25, 'MgO': 0.25, 'FeO': 0.25}

Returns [Pa.s] dynamic viscosity

The **state parameter contains the keyword argument(s) specified above that are used to describe the state of the material.

A model that describes the variation in the dynamic viscosity of liquid slag as a function of temperature and composition expressed in mass fraction.

```
calculate ( **state)
```

Calculate dynamic viscosity at the specified temperature and composition:

Parameters

- $\mathbf{T} [K]$ temperature
- **y** [mass fraction] composition dictionary, e.g. {'SiO2': 0.25, 'CaO': 0.25, 'MgO': 0.25, 'FeO': 0.25}

Returns [Pa.s] dynamic viscosity

The **state parameter contains the keyword argument(s) specified above that are used to describe the state of the material.

Module contents

This package provides tools to calculate material physical properties.

auxi.tools.transportphenomena package

Submodules

auxi.tools.transportphenomena.dimensionlessquantities module

This module provides functions to calculate dimensionless quantities used when doing transport phenomena calculations.

```
auxi.tools.transportphenomena.dimensionlessquantities. \mathbf{Gr} ( L: float, Ts: float, Tf: float, beta: float, nu: float, t: float, t:
```

Calculate the Grashof number.

Parameters

- **L** [m] heat transfer surface characteristic length.
- **Ts** [K] heat transfer surface temperature.
- **Tf** [K] bulk fluid temperature.
- beta [1/K] fluid coefficient of thermal expansion.
- nu [m2/s] fluid kinematic viscosity.

Returns float

$$Gr = \frac{g\beta(Ts - Tinf)L^3}{\nu^2}$$

Characteristic dimensions:

• vertical plate: vertical length

• pipe: diameter

• bluff body: diameter

Parameters

- L [m] heat transfer surface characteristic length.
- **h** [W/K/m2] convective heat transfer coefficient.
- **k** [W/K/m] fluid thermal conductivity.

Returns float

```
auxi.tools.transportphenomena.dimensionlessquantities. Pr ( \it nu: float, \it al-pha: float) \rightarrow float
```

Parameters

Calculate the Prandtl number.

- nu [m2/s] fluid kinematic viscosity / momentum diffusivity.
- alpha [m2/s] fluid thermal diffusivity.

Returns float

```
auxi.tools.transportphenomena.dimensionlessquantities. Ra ( L: float, Ts: float, Tf: float, alpha: float, beta: float, nu: float) \rightarrow float
```

Calculate the Grashof number.

Parameters

- L [m] heat transfer surface characteristic length.
- Ts [K] heat transfer surface temperature.
- **Tf** [K] bulk fluid temperature.
- alpha [m2/s] fluid thermal diffusivity.
- beta [1/K] fluid coefficient of thermal expansion.
- nu [m2/s] fluid kinematic viscosity.

Returns float

Ra = Gr*Pr

Characteristic dimensions:

• vertical plate: vertical length

• pipe: diameter

• bluff body: diameter

```
auxi.tools.transportphenomena.dimensionlessquantities. Re ( L: float, v: float, nu: float) \rightarrow float
```

Calculate the Reynolds number.

Parameters

- L [m] surface characteristic length.
- $\mathbf{v} [\text{m/s}]$ fluid velocity relative to the object.
- nu [m2/s] fluid kinematic viscosity.

Returns float

```
auxi.tools.transportphenomena.dimensionlessquantities. Sh ( L: float, h: float, D: float) \rightarrow float
```

Calculate the Sherwood number.

Parameters

- **L** [m] mass transfer surface characteristic length.
- h [m/s] mass transfer coefficient.

• **D** – [m2/s] fluid mass diffusivity.

Returns float

Module contents

This package provides tools to do calculations related to heat transfer, mass transfer, and fluid flow.

Submodules

auxi.tools.physicalconstants module

This module provides a set of physical constants that are used frequently.

```
auxi.tools.physicalconstants. \mathbf{F} = 96485.3365 [C·mol-1] Faraday constant
```

auxi.tools.physicalconstants. G = 6.67408e-11 [m3.kg-1.s-2] Newtonian constant of gravitation

```
auxi.tools.physicalconstants. N_A = 6.02214129e+23 [mol-1] Avogadro's number
```

```
auxi.tools.physicalconstants. R = 8.3144621 [J.K-1.mol-1] gas constant
```

```
auxi.tools.physicalconstants. c = 299792458 [m·s-1] speed of light in vacuum
```

```
auxi.tools.physicalconstants. g = 9.80665 [m.s-2] standard acceleration of gravity
```

```
auxi.tools.physicalconstants. h = 6.62607004e-34 [J.s] Planck constant
```

```
auxi.tools.physicalconstants. h_bar = 1.0545718e-34 [J.s] reduced Planck constant (h/(2pi))
```

```
auxi.tools.physicalconstants. k_B = 1.3806488e-23 [J.K-1] Boltzmann constant
```

```
auxi.tools.physicalconstants.m_u = 1.660538921e-27 [kg] atomic mass constant
```

Module contents

This package provides tools to make common tasks quick, easy and accurate.

4.1.2 Submodules

auxi.tests module

This module runs all the tests of the auxi package at once.

4.1.3 Module contents

A toolkit to help metallurgical process engineers to rapidly do day-to-day calculations.

auxi is a toolkit to help metallurgical process engineers with their day-to-day tasks. Many of the calculations that we do require things like molar masses, conversion of one compound to another using stoichiometry, heat transfer calculations, mass balances, energy balances, etc. It is usually quite time consuming to get started with these calculations in a tool like Excel. auxi aims to save you time by making many of these calculations available from within python.

We hope that auxi will help you spend less time focusing on searching for formulas and data, and setting up calculations, and more on thinking about the problems that you need to solve with these calculations. Enjoy!

If you need more information about how to use auxi, please refer the documentation at https://auxi.readthedocs.org/en/latest/.

CHAPTER

FIVE

INDICES AND TABLES

- genindex
- modindex
- search

PYTHON MODULE INDEX

```
a
                                       92
                                 auxi.tests, 116
auxi, 116
                                 auxi.tools, 115
auxi.core, 53
                                 auxi.tools.chemistry, 102
auxi.core.helpers, 49
                                 auxi.tools.chemistry.stoichiometry_test,
auxi.core.helpers_test,49
auxi.core.objects, 50
                                 auxi.tools.chemistry.thermochemistry_test
auxi.core.objects_test,50
auxi.core.reporting, 51
                                 auxi.tools.materialphysicalproperties,
auxi.core.reporting_test,51
auxi.core.time, 52
                                 auxi.tools.materialphysicalproperties.com
auxi.core.time_test,53
auxi.modelling, 93
                                 auxi.tools.materialphysicalproperties.cor
auxi.modelling.business, 64
auxi.modelling.business.basic_test,
                                 auxi.tools.materialphysicalproperties.gas
     56
                                       105
auxi.modelling.business.models_test,
                                 auxi.tools.materialphysicalproperties.ide
auxi.modelling.business.structure_test^{105}
                                 auxi.tools.materialphysicalproperties.ide
                                       107
auxi.modelling.financial,72
auxi.modelling.financial.des_testuxi.tools.materialphysicalproperties.lig
                                       108
auxi.modelling.financial.reportiagxieteols.materialphysicalproperties.pol
                                 auxi.tools.materialphysicalproperties.pol
auxi.modelling.process, 93
                                       109
auxi.modelling.process.materials,
                                 auxi.tools.materialphysicalproperties.sla
auxi.modelling.process.materials.chem_test,
                                 auxi.tools.physicalconstants,
auxi.modelling.process.materials.core, 115
                                 auxi.tools.transportphenomena,
auxi.modelling.process.materials.core_16st,
                                 auxi.tools.transportphenomena.dimensionle
auxi.modelling.process.materials.psd_test,
auxi.modelling.process.materials.slurry_test,
     87
auxi.modelling.process.materials.thermo_test,
```

Α			amount	_fractions()	(in	module
AccountType	(class	in		auxi.tools.che	emistry.stoichi	ometry),
• •	odelling.financial.des), 64			94		
Activity	(class	in	amounts		(in	module
auxi.m	odelling.business.structure)),			mistry.stoichi	ometry),
59				94		
ActivityUnitTes	· ·	in		lmostEqual()		
	odelling.business.structure	_test)	,	,	ng.process.ma	nterials.chem_test.ChemN
62			· · · · · · · · · · · · · · · ·	method), 77		
	auxi.modelling.business.str	ructur	eicethpe	mostEqual()		iomotime to at Chaigh France
method					emistry.stoicn	iometry_test.StoichFunct
	ixi.modelling.process.mate	rials.c	chem.Ma	most Equal()		
method	1), 74		asserta	mostEqual()	emistry therm	ochemistry test Thermol
add_assay() (au	exi.modelling.process.mate	rıals.p	osd.Mate	method), 102	emisu y.memi	ochemistry_test.Thermor
method	1), 80		1255et 1 (21		financial des /	AccountTyne
add_assay() (au	exi.modelling.process.mate	riais.s	Srurry: IVI	attribute), 64	mianeiar.acs.z	recount type
method	ixi.modelling.process.mate	riole t	ıATm(in)	, ,	modelling.fina	ncial.des).
method		11a18.1	11671110:10	64		,
add_incompatib			auxi (m	odule), 116		
	nodelling.process.materials	chen	•	* * * * * * * * * * * * * * * * * * * *	3 ckageUnitTes	ter
method		.011011		e.helpers (mo		
add_incompatib			auxi.com	e.helpers_test	(module), 49	
(auxi.n	nodelling.process.materials	.thern	nauxi.go	hehicMatee	ส์เคลื่อใหล้ใน	tTester
method	1), 92		auxi.com	e.objects_test	(module), 50	
add_to() (auxi.r	nodelling.process.materials	s.cher	n! Mater i	apaprija g (n	nodule), 51	
method	1), 75		aux1.com	e.reporting_te	est (module), 5	$\delta 1$
add_to() (auxi.r	nodelling.process.materials	s.psd.	Myxiral	p atimeg(modul	le), 52	
method	d), 81		aux1.com	e.time_test (n	10dule), 53	
alpha() (auxi.m	odelling.process.materials.	core.l	Matei ino	delling (modu	ıle), 93	
method	1), 78		aux1.mo	delling.busine	ess (module), (
amount (auxi.m	odelling.business.basic.Ba	sicLo	allactive	delling.busine	ess.basic (mod	ule), 53
attribut	re), 55		aux1.mo	delling.busine	ess.basic_test	(module),
	odelling.process.materials.	thern	no.Mater	ialPackage	1.1.7	1.1.
attribut	, · ·			delling.busine	,	
amount()	`	dule	aux1.mo	delling.busine	ess.models_tes	st (mod-
	ols.chemistry.stoichiometry	y),	01111	ule), 58	aa atmaataa	(madula)
94			aux1.III0	delling.busine	ss.structure	(module),

59	(module), 103	
auxi.modelling.business.structure_test (mod-	auxi.tools.materialphysicalproperties.core_test	Ļ
ule), 62	(module), 104	
auxi.modelling.financial (module), 72	auxi.tools.materialphysicalproperties.gases	
auxi.modelling.financial.des (module), 64	(module), 105	
auxi.modelling.financial.des_test (module),	auxi.tools.materialphysicalproperties.idealgas	
69	(module), 105	
auxi.modelling.financial.reporting (module),	auxi.tools.materialphysicalproperties.idealgas_	_test
70	(module), 107	
auxi.modelling.financial.reporting_test (mod-	auxi.tools.materialphysicalproperties.liquids	
ule), 72	(module), 108	
auxi.modelling.process (module), 93	auxi.tools.materialphysicalproperties.polynom	ial
auxi.modelling.process.materials (module),	(module), 108	
93	auxi.tools.materialphysicalproperties.polynom	ial_test
auxi.modelling.process.materials.chem (mod-	(module), 109	
ule), 73	auxi.tools.materialphysicalproperties.slags	
auxi.modelling.process.materials.chem_test	(module), 110	
(module), 76	auxi.tools.physicalconstants (module), 115	
auxi.modelling.process.materials.core (mod-	auxi.tools.transportphenomena (module), 115	
ule), 77	auxi.tools.transportphenomena.dimensionlesso	quantities
auxi.modelling.process.materials.core_test	(module), 112	
(module), 79	В	
auxi.modelling.process.materials.psd (mod-		
ule), 79	balance_sheet (auxi.modelling.financial.report	ing.ReportType
auxi.modelling.process.materials.psd_test	attribute), 71	
(module), 82	balance_sheet()	
auxi.modelling.process.materials.slurry (mod-	(auxi.modelling.financial.des.General	Ledger
ule), 83	method), 65	
auxi.modelling.process.materials.slurry_test	BalanceSheet (class in	
(module), 87	auxi.modelling.financial.reporting),	
auxi.modelling.process.materials.thermo	70	
(module), 88	BasicActivity (class in	
auxi.modelling.process.materials.thermo_test	auxi.modelling.business.basic),	
(module), 92	53	
auxi.tests (module), 116	BasicActivityUnitTester (class in	
auxi.tools (module), 115	auxi.modelling.business.basic_test),	
auxi.tools.chemistry (module), 102	56	
auxi.tools.chemistry.stoichiometry (module),	BasicLoanActivity (class in	
94	auxi.modelling.business.basic),	
auxi.tools.chemistry.stoichiometry_test (mod-	54	
ule), 97	BasicLoanActivityUnitTester (class in	
auxi.tools.chemistry.thermochemistry (mod-	auxi.modelling.business.basic_test),	
ule), 98	56	
auxi.tools.chemistry.thermochemistry_test	beta() (auxi.modelling.process.materials.core.l	Material
(module), 102	method), 78	
auxi.tools.materialphysicalproperties (mod-	BetaT (class in	
ule), 112	auxi.tools.materialphysicalproperties.i	idealgas),
auxi.tools.materialphysicalproperties.core	105	

```
BetaTTester
                                                                              (class
                                                                                                                                       in clone() (auxi.modelling.process.materials.psd.MaterialPackag
                          auxi.tools.materialphysicalproperties.idealgas_textothod), 81
                                                                                                                                                    clone() (auxi.modelling.process.materials.slurry.MaterialPack
                          107
                                                                                                                                                                             method), 85
                                                                                                                                                    clone() (auxi.modelling.process.materials.thermo.MaterialPac
c (in module auxi.tools.physicalconstants),
                                                                                                                                                                             method), 90
                          115
                                                                                                                                                    Component
                                                                                                                                                                                                                                 (class
                                                                                                                                                                                                                                                                                            in
calculate() (auxi.tools.materialphysicalproperties.core.Madel.modelling.business.structure),
                         method), 104
calculate() (auxi.tools.materialphysicalpropertie: Cidente and Tester
                                                                                                                                                                                                                                                 (class
                                                                                                                                                                                                                                                                                            in
                         method), 105
                                                                                                                                                                             auxi.modelling.business.structure_test),
calculate() (auxi.tools.materialphysicalproperties.idealgas@RhoT
                         method), 106
                                                                                                                                                    Compound
                                                                                                                                                                                                                                (class
calculate() (auxi.tools.materialphysicalproperties.idealgaauRhoTels.chemistry.thermochemistry),
                         method), 106
calculate() (auxi.tools.materialphysicalproperties; idealgas RhoTPlx
                         method), 106
                                                                                                                                                                             (auxi.modelling.process.materials.thermo.Material
calculate() (auxi.tools.materialphysicalproperties.polynomialModelT
                         method), 109
                                                                                                                                                    compounds (auxi.modelling.process.materials.thermo.Materials.thermo.Materials.thermo.Materials.thermo.Materials.thermo.Materials.thermo.Materials.thermo.Materials.thermo.Materials.thermo.Materials.thermo.Materials.thermo.Materials.thermo.Materials.thermo.Materials.thermo.Materials.thermo.Materials.thermo.Materials.thermo.Materials.thermo.Materials.thermo.Materials.thermo.Materials.thermo.Materials.thermo.Materials.thermo.Materials.thermo.Materials.thermo.Materials.thermo.Materials.thermo.Materials.thermo.Materials.thermo.Materials.thermo.Materials.thermo.Materials.thermo.Materials.thermo.Materials.thermo.Materials.thermo.Materials.thermo.Materials.thermo.Materials.thermo.Materials.thermo.Materials.thermo.Materials.thermo.Materials.thermo.Materials.thermo.Materials.thermo.Materials.thermo.Materials.thermo.Materials.thermo.Materials.thermo.Materials.thermo.Materials.thermo.Materials.thermo.Materials.thermo.Materials.thermo.Materials.thermo.Materials.thermo.Materials.thermo.Materials.thermo.Materials.thermo.Materials.thermo.Materials.thermo.Materials.thermo.Materials.thermo.Materials.thermo.Materials.thermo.Materials.thermo.Materials.thermo.Materials.thermo.Materials.thermo.Materials.thermo.Materials.thermo.Materials.thermo.Materials.thermo.Materials.thermo.Materials.thermo.Materials.thermo.Materials.thermo.Materials.thermo.Materials.thermo.Materials.thermo.Materials.thermo.Materials.thermo.Materials.thermo.Materials.thermo.Materials.thermo.Materials.thermo.Materials.thermo.Materials.thermo.Materials.thermo.Materials.thermo.Materials.thermo.Materials.thermo.Materials.thermo.Materials.thermo.Materials.thermo.Materials.thermo.Materials.thermo.Materials.thermo.Materials.thermo.Materials.thermo.Materials.thermo.Materials.thermo.Materials.thermo.Materials.thermo.Materials.thermo.Materials.thermo.Materials.thermo.Materials.thermo.Materials.thermo.Materials.thermo.Materials.thermo.Materials.thermo.Materials.thermo.Materials.thermo.Materials.thermo.Materials.thermo.Materials.thermo.Materials.thermo.Materials.thermo.Materi
calculate() (auxi.tools.materialphysicalproperties.slags.RiboutdWiscosityTx
                         method), 110
                                                                                                                                                    convert_compound()
                                                                                                                                                                                                                                                                          module
                                                                                                                                                                                                                                           (in
calculate() (auxi.tools.materialphysicalproperties.slags.RaboudVilcohityhty.stoichiometry),
                         method), 110
calculate() (auxi.tools.materialphysicalproperties: slags: letbain: Viscosity Tx
                         method), 111
                                                                                                                                                                              (auxi.modelling.process.materials.thermo.Material
calculate() (auxi.tools.materialphysicalproperties.slags.UzhainWiscosityTy
                         method), 111
                                                                                                                                                    Cp() (auxi.modelling.process.materials.core.Material
cash_flow (auxi.modelling.financial.reporting.ReportTypmethod), 78
                         attribute), 71
                                                                                                                                                    Cp() (auxi.tools.chemistry.thermochemistry.Compound
ChemMaterialPackageUnitTester (class
                                                                                                                                                                              method), 98
                         auxi.modelling.process.materials.chem\_ \textit{(auxi.tools.chemistry.thermochemistry.CpRecord)} (auxi.tools.chemistry.thermochemistry.CpRecord) (auxi.tools.chemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemis
                                                                                                                                                                             method), 99
ChemMaterialUnitTester
                                                                                                  (class
                                                                                                                                        in Cp() (auxi.tools.chemistry.thermochemistry.Phase
                         auxi.modelling.process.materials.chem_test),
                                                                                                                                                                             method), 100
                                                                                                                                                    Cp()
                                                                                                                                                                                                                                                                          module
clear() (auxi.modelling.process.materials.chem.MaterialPackagols.chemistry.thermochemistry),
                         method), 75
clear() (auxi.modelling.process.materials.psd.Materiale.psd.Materiale.psd.Materiale.psd.Materiale.psd.Materiale.psd.Materiale.psd.Materiale.psd.Materiale.psd.Materiale.psd.Materiale.psd.Materiale.psd.Materiale.psd.Materiale.psd.Materiale.psd.Materiale.psd.Materiale.psd.Materiale.psd.Materiale.psd.Materiale.psd.Materiale.psd.Materiale.psd.Materiale.psd.Materiale.psd.Materiale.psd.Materiale.psd.Materiale.psd.Materiale.psd.Materiale.psd.Materiale.psd.Materiale.psd.Materiale.psd.Materiale.psd.Materiale.psd.Materiale.psd.Materiale.psd.Materiale.psd.Materiale.psd.Materiale.psd.Materiale.psd.Materiale.psd.Materiale.psd.Materiale.psd.Materiale.psd.Materiale.psd.Materiale.psd.Materiale.psd.Materiale.psd.Materiale.psd.Materiale.psd.Materiale.psd.Materiale.psd.Materiale.psd.Materiale.psd.Materiale.psd.Materiale.psd.Materiale.psd.Materiale.psd.Materiale.psd.Materiale.psd.Materiale.psd.Materiale.psd.Materiale.psd.Materiale.psd.Materiale.psd.Materiale.psd.Materiale.psd.Materiale.psd.Materiale.psd.Materiale.psd.Materiale.psd.Materiale.psd.Materiale.psd.Materiale.psd.Materiale.psd.Materiale.psd.Materiale.psd.Materiale.psd.Materiale.psd.Materiale.psd.Materiale.psd.Materiale.psd.Materiale.psd.Materiale.psd.Materiale.psd.Materiale.psd.Materiale.psd.Materiale.psd.Materiale.psd.Materiale.psd.Materiale.psd.Materiale.psd.Materiale.psd.Materiale.psd.Materiale.psd.Materiale.psd.Materiale.psd.Materiale.psd.Materiale.psd.Materiale.psd.Materiale.psd.Materiale.psd.Materiale.psd.Materiale.psd.Materiale.psd.Materiale.psd.Materiale.psd.Materiale.psd.Materiale.psd.Materiale.psd.Materiale.psd.Materiale.psd.Materiale.psd.Materiale.psd.Materiale.psd.Materiale.psd.Materiale.psd.Materiale.psd.Materiale.psd.Materiale.psd.Materiale.psd.Materiale.psd.Materiale.psd.Materiale.psd.Materiale.psd.Materiale.psd.Materiale.psd.Materiale.psd.Materiale.psd.Materiale.psd.Materiale.psd.Materiale.psd.Materiale.psd.Materiale.psd.Materiale.psd.Materiale.psd.Materiale.psd.Materiale.psd.Materiale.psd.Materiale.psd.Materiale.psd.Materiale.psd.Materiale.psd.Materiale.ps
                                                                                                                                                                                                                               (class
                         method), 81
                                                                                                                                                                              auxi.tools.chemistry.thermochemistry),
clear() (auxi.modelling.process.materials.slurry.MaterialPackage
                         method), 85
                                                                                                                                                    create() (auxi.tools.materialphysicalproperties.polynomial.Po
clear() (auxi.modelling.process.materials.thermo.MaterialPackage 109
                         method), 90
                                                                                                                                                    create_account()
Clock (class in auxi.core.time), 52
                                                                                                                                                                              (auxi.modelling.financial.des.GeneralLedgerAccount
ClockUnitTester (class in auxi.core.time_test),
                                                                                                                                                                             method), 66
                                                                                                                                                    create_component()
clone() (auxi.modelling.process.materials.chem.MaterialPackaggodelling.business.structure.Component
                         method), 75
                                                                                                                                                                             method), 60
```

```
create_component()
                                                                                                         attribute), 89
               (auxi.modelling.business.structure.EntitDHref (auxi.tools.chemistry.thermochemistry.Phase
               method), 61
                                                                                                        attribute), 100
create_empty_assay()
               (auxi.modelling.process.materials.chem.Material
                                                                                                                                     (class
                                                                                                                                                                           in
               method), 74
                                                                                                         auxi.tools.chemistry.stoichiometry),
create_empty_assay()
               (auxi.modelling.process.materials.psd.Material 94
                                                                                         element_mass_fraction()
                                                                                                                                                  (in
               method), 80
                                                                                                         auxi.tools.chemistry.stoichiometry),
create_empty_assay()
               (auxi.modelling.process.materials.slurry.Material<sup>95</sup>
                                                                                         element_mass_fractions()
                                                                                                                                                   (in
                                                                                                                                                                 module
               method), 84
                                                                                                         auxi.tools.chemistry.stoichiometry),
create_empty_assay()
               (auxi.modelling.process.materials.thermo.Material
                                                                                         elements()
                                                                                                                                                                 module
                                                                                                                                    (in
               method), 89
                                                                                                         auxi.tools.chemistry.stoichiometry),
create_entity()
               (auxi.modelling.business.models.TimeBasedModel
                                                                                                                                   (class
                                                                                                                                                                           in
                                                                                                         auxi.modelling.business.structure),
create_package()
               (auxi.modelling.process.materials.chem.Materia<sup>61</sup>
                                                                                         EntityUnitTester
                                                                                                                                             (class
                                                                                                                                                                           in
               method), 74
                                                                                                         auxi.modelling.business.structure_test),
create_package()
               (auxi.modelling.process.materials.psd.Material 63
                                                                                         equity (auxi.modelling.financial.des.AccountType
               method), 80
                                                                                                         attribute), 64
create_package()
               (auxi.modelling.process.materials.slurry \verb| Mansea (auxi.modelling.financial.des. Account Type) | (auxi.modelling.process.materials.slurry \verb| Mansea (auxi.modelling.financial.des. Account Type) | (auxi.modelling
                                                                                                         attribute), 64
               method), 85
                                                                                         extract() (auxi.modelling.process.materials.chem.MaterialPac
create_package()
               (auxi.modelling.process.materials.thermo.Mater {\tt Mater} {\tt method}),\,75
                                                                                         extract() (auxi.modelling.process.materials.psd.MaterialPacka
               method), 89
                                                                                                         method), 81
create_template()
               (auxi.tools.materialphysicalproperties.cort.portal auxi.modelling.process.materials.slurry.MaterialPac
                                                                                                         method), 85
               method), 103
                                                                                         extract() (auxi.modelling.process.materials.thermo.MaterialP
create_transaction()
                                                                                                         method), 91
               (auxi.modelling.financial.des.GeneralLedger
               method), 65
                                                                                          F
           (auxi.core.reporting.ReportFormat
                                                                                         F (in module auxi.tools.physicalconstants),
               tribute), 51
D
                                                                                         formula (auxi.tools.chemistry.thermochemistry.Compound
                                                                                                         attribute), 99
DataSet
                                           (class
                                                                                  in
               auxi.tools.materialphysicalproperties.com,
                                                                                         G (in module auxi.tools.physicalconstants),
DataSetTester
                                                 (class
               auxi.tools.materialphysicalproperties.core_test), 115
104 g (in module auxi.tools.physicalconstants),
                                                                                                         115
day (auxi.core.time.TimePeriod attribute), 52
\label{eq:gamma} description~(auxi.modelling.process.materials.thermo.Material)} G()~(auxi.tools.chemistry.thermochemistry.Compound)
                                                                                                         method), 98
```

```
G() (auxi.tools.chemistry.thermochemistry.Phaseet_assay_total()
                    method), 100
                                                                                                                                        (auxi.modelling.process.materials.slurry.Material
                                                                                                                                        method), 85
G()
                                                                                             module
                    auxi.tools.chemistry.thermochemistry), get_assay_total()
                                                                                                                                        (auxi.modelling.process.materials.thermo.Material
gas (auxi.tools.materialphysicalproperties.core.StateOfMattethod), 89
                    attribute), 104
                                                                                                                    get_child_account()
gas_plasma (auxi.tools.materialphysicalproperties.core.StateOfMatebling.financial.des.GeneralLedgerAccount
                    attribute), 104
                                                                                                                                        method), 66
GeneralLedger
                                                                                                                   get_component()
                                                                (class
                    auxi.modelling.financial.des), 64
                                                                                                                                        (auxi.modelling.business.structure.Component\\
GeneralLedgerAccount
                                                                                                                                        method), 60
                                                                           (class
                                                                                                          in
                    auxi.modelling.financial.des), 66
                                                                                                                    get_compound_amount()
GeneralLedgerAccountUnitTester (class
                                                                                                                                        (auxi.modelling.process.materials.thermo.MaterialPa
                                                                                                        in
                    auxi.modelling.financial.des_test), 69
                                                                                                                                        method), 91
                                                                           (class
                                                                                                                    get_compound_amounts()
GeneralLedgerStructure
                                                                                                          in
                    auxi.modelling.financial.des), 67
                                                                                                                                        (auxi.modelling.process.materials.thermo.Material Parameters and all process.materials.thermo.Material Parameters and all process.materials.thermo.Materials.thermo.Materials.thermo.Materials.thermo.Materials.thermo.Materials.thermo.Materials.thermo.Materials.thermo.Materials.thermo.Materials.thermo.Materials.thermo.Materials.thermo.Materials.thermo.Materials.thermo.Materials.thermo.Materials.thermo.Materials.thermo.Materials.thermo.Materials.thermo.Materials.thermo.Materials.thermo.Materials.thermo.Materials.thermo.Materials.thermo.Materials.thermo.Materials.thermo.Materials.thermo.Materials.thermo.Materials.thermo.Materials.thermo.Materials.thermo.Materials.thermo.Materials.thermo.Materials.thermo.Materials.thermo.Materials.thermo.Materials.thermo.Materials.thermo.Materials.thermo.Materials.thermo.Materials.thermo.Materials.thermo.Materials.thermo.Materials.thermo.Materials.thermo.Materials.thermo.Materials.thermo.Materials.thermo.Materials.thermo.Materials.thermo.Materials.thermo.Materials.thermo.Materials.thermo.Materials.thermo.Materials.thermo.Materials.thermo.Materials.thermo.Materials.thermo.Materials.thermo.Materials.thermo.Materials.thermo.Materials.thermo.Materials.thermo.Materials.thermo.Materials.thermo.Materials.thermo.Materials.thermo.Materials.thermo.Materials.thermo.Materials.thermo.Materials.thermo.Materials.thermo.M
GeneralLedgerStructure
                                                                                                                                        method), 91
                                                                                                          in
                    auxi.modelling.financial.reporting),
                                                                                                                    get_compound_index()
                                                                                                                                        (auxi.modelling.process.materials.chem.Material\\
GeneralLedgerStructureUnitTester (class in
                                                                                                                                        method), 74
                                                                                                                    get_compound_index()
                    auxi.modelling.financial.des_test), 69
GeneralLedgerStructureUnitTester (class in
                                                                                                                                        (auxi.modelling.process.materials.thermo.Material
                    auxi.modelling.financial.reporting_test),
                                                                                                                                        method), 89
                    72
                                                                                                                    get_compound_mass()
GeneralLedgerUnitTester
                                                                                                                                        (auxi.modelling.process.materials.chem.MaterialPac
                                                                             (class
                    auxi.modelling.financial.des_test), 69
                                                                                                                                        method), 75
get\_account() \ (auxi.modelling.financial.des.Gen \ \underline{\textbf{get}} \underline{\textbf{IlccdgpoStrd}}\underline{\textbf{ctmas}}s()
                    method), 67
                                                                                                                                        (auxi.modelling.process.materials.thermo.Material Parameters and approximate process.material parameters and approximate process.material parameters and approximate process. The process of the proces
get_account_decendants()
                                                                                                                                        method), 91
                    (auxi.modelling.financial.des.General Legter Strup our ed\_mass\_fraction()
                                                                                                                                       (auxi.modelling.process.materials.chem.MaterialPac
get_activity() (auxi.modelling.business.structure.Compomenthod), 75
                                                                                                                    get_datafile_references()
                    method), 60
                                                                                                                                                                                             (in
                                                                                                                                                                                                                 module
get_assay() (auxi.modelling.process.materials.chem.MaterialRockaghemistry.thermochemistry),
                    method), 75
get_assay() (auxi.modelling.process.materials.psgeMdtteialParchageule auxi.core.helpers), 49
                                                                                                                   get_datetime()
                                                                                                                                                                               (auxi.core.time.Clock
                    method), 81
get_assay() (auxi.modelling.process.materials.slurry.Materialheat)kage
                                                                                                                    get_datetime_at_period_ix()
                    method), 86
get_assay() (auxi.modelling.process.materials.thermo.MaterialParcktigee.Clock
                                                                                                                                                                                                              method),
                    method), 91
                                                                                                                    get_density() (auxi.modelling.process.materials.slurry.Materi
get_assay_total()
                    (auxi.modelling.process.materials.chem.Materiahethod), 86
                    method), 74
                                                                                                                    get_element_mass()
                                                                                                                                        (auxi.modelling.process.materials.chem.MaterialPac
get_assay_total()
                    (auxi.modelling.process.materials.psd.Material method), 76
                                                                                                                    get_element_mass()
                    method), 80
```

```
(auxi.modelling.process.materials.thermo.MaterialRhoka)g82
                    method), 91
                                                                                                                     get_size_class_mass()
                                                                                                                                          (auxi.modelling.process.materials.slurry.MaterialPac
get_element_mass_dictionary()
                    (auxi.modelling.process.materials.chem.MaterialRattkade 86
                    method), 76
                                                                                                                     get_size_class_mass_fraction()
                                                                                                                                          (auxi.modelling.process.materials.psd.MaterialPacka
get_element_mass_dictionary()
                    (auxi.modelling.process.materials.thermo.Material?hoklag82
                    method), 91
                                                                                                                     get_size_class_mass_fraction()
                                                                                                                                          (auxi.modelling.process.materials.slurry.MaterialPac
get_element_masses()
                    (auxi.modelling.process.materials.chem.MaterialRettkad); 86
                    method), 76
                                                                                                                     get_solid_mass()
get_element_masses()
                                                                                                                                          (auxi.modelling.process.materials.slurry.MaterialPac
                    (auxi.modelling.process.materials.thermo.MaterialRhokl)g86
                    method), 92
                                                                                                                     get_volume() (auxi.modelling.process.materials.slurry.Materi
get_mass() (auxi.modelling.process.materials.chem.MaterialPadkage
                    method), 76
                                                                                                                     get_volume_fraction_solids()
get_mass() (auxi.modelling.process.materials.psd.Materi@alPackagedelling.process.materials.slurry.MaterialPac
                                                                                                                                          method), 86
get_mass() (auxi.modelling.process.materials.slu@ry)MaterialPackage(in
                                                                                                                                                                                                                    module
                    method), 86
                                                                                                                                          auxi.tools.transportphenomena.dimensionlessquantit
get_mass_fraction_solids()
                   (auxi.modelling.process.materials.slurry \underline{\textbf{\textit{M}}} aterial Package
                    method), 86
                                                                                                                   H (auxi.modelling.process.materials.thermo.MaterialPackage
get_path_relative_to_module()
                                                                             (in module
                                                                                                                                          attribute), 90
                    auxi.core.helpers), 49
                                                                                                                     h (in module auxi.tools.physicalconstants),
get_phase_list()
                   (auxi.tools.chemistry.thermochemistry.Compound 5
                                                                                                                      H() (auxi.tools.chemistry.thermochemistry.Compound
                    method), 99
                                                                                                                                          method), 98
get_reference()
                   (auxi.tools.chemistry.thermochemistry. Compound tools.chemistry.thermochemistry. CpRecord tools.chemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.thermochemistry.th
                                                                                                                                          method), 99
                    method), 99
                                                                                                                      H() (auxi.tools.chemistry.thermochemistry.Phase
get_referenced_accounts()
                                                                                                                                          method), 101
                    (auxi.modelling.business.basic.BasicActivity
                    method), 54
                                                                                                                                          auxi.tools.chemistry.thermochemistry),
get_referenced_accounts()
                   (auxi.modelling.business.basic.BasicLoanActivity)0
                                                                                                                      h bar
                                                                                                                                                                                                                    module
                                                                                                                                                                         (in
                    method), 55
                                                                                                                                          auxi.tools.physicalconstants), 115
get_referenced_accouts()
                   (auxi.modelling.business.structure.Activity \cite{thetapersUnitTester}
                                                                                                                                                                                           (class
                                                                                                                                                                                                                                  in
                                                                                                                                          auxi.core.helpers_test), 49
                    method), 59
                                                                                                                     hour (auxi.core.time.TimePeriod attribute), 52
get_size_class_index()
                    (auxi.modelling.process.materials.psd.Material
                    method), 80
                                                                                                                     income statement
get_size_class_index()
                    (auxi.modelling.process.materials.slurry.Material (auxi.modelling.financial.reporting.ReportType
                                                                                                                                           attribute), 71
                    method), 85
                                                                                                                      income_statement()
get_size_class_mass()
                   (auxi.modelling.process.materials.psd.MaterialPackage, \\ (auxi.modelling.financial.des.GeneralLedger, \\ (auxi.modelling.process.materials.psd.MaterialPackage, \\ (auxi.modelling.financial.des.GeneralLedger, \\ (auxi.modelling.des.GeneralLedger, \\ (auxi.modelling.financial.des.GeneralLedger, \\ (auxi.modelli
                                                                                                                                          method), 65
```

Income	Statement	(class	in	Materia	al		(class		in
	auxi.modelling.	financial.repo	orting),		auxi.	.modelling	g.process.r	naterials.cl	hem),
	70				73				
interest_	_rate (auxi.mode	lling.business	s.basic.Bas	siMateuri <i>a</i>	Activi	ty	(class		in
	attribute), 55					.modelling	g.process.i	naterials.c	ore),
K				Materia	77 al		(class		in
k() (aux	i.modelling.proc	ess.materials	.core.Mate			modelling	`	naterials.p	
	method), 78				79	inio de ining	5.p100055.1	naterials.p	34),
k B (in	module auxi.too	ols.physicalco	onstants),	Materia			(class		in
_ \	115	1 7	,,	1,14,0110		modelling	`	naterials.sl	
					83	inio de ining	5.process.r	1141011415151	,,
L				Materia			(class		in
latex (auxi.core.reporti	ing.ReportFor	rmat at-	William		modelling	`	naterials.th	
	tribute), 51				88	inio de ining	5.process.r	materials.ti	(CIIIIO),
liability	(auxi.modelling	financial.des	.AccountT	W Steria	alPack	age	(class	3	in
	attribute), 64			-1-1410110			`	naterials.cl	
liquid (a	auxi.tools.materi	alphysicalpro	perties.com	re.StateC	Of Mat	ter	5.p100055.1	iiateriaiste.	,,
_	attribute), 104			Materia	ılPack	age	(class	3	in
liquid_g	gas_plasma			1,10000110		_	`	naterials.p	
	(auxi.tools.mate	erialphysicalp	roperties.c	ore.Stat			5.p100055.1	naterials.p	34),
	attribute), 104			Materia	alPack	age	(class	S	in
liquid_p	olasma (auxi.tool	ls.materialphy	sicalprope	erties.co	re Sta	te Of Matte	erprocess.i	naterials.sl	
	attribute), 104				85		5·I		- 3 //
list_con	npounds()	(in	module	Materia	alPack	age	(class	8	in
	auxi.tools.chem	istry.thermoc	hemistry),				`	naterials.th	
	101				90		5·I		,,
load_da	ıta_auxi()	(in	module	Materia		er	(class		in
	auxi.tools.chem	istry.thermoc	hemistry),				•	naterials.c	
	101				79		5·I		
load_da	ita_factsage()	(in	module	matplot	tlib (a	auxi.core.i	reporting.I	ReportForm	nat
	auxi.tools.chem	istry.thermoc	hemistry),	1		oute), 51	1 6	1	
	101			millised		, ,	e.time.Tin	nePeriod	at-
М						te), 52			
				minute		* '	e.TimePeri	od attribut	te),
m_u (in	module auxi.too	ols.physicalco	onstants),		52				,
	115			mm (au	xi.too	ls.materia	lphysical	properties.i	dealgas.RhoT
mass (a	uxi.modelling.pr	ocess.materia	ıls.thermo.	Materia	IPack: attrit	age oute), 106		-	
	attribute), 92						lphysicalp	properties.i	dealgas.RhoTP
mass()	(in		module		attrib	oute), 106		_	_
	auxi.tools.chem	istry.stoichio	metry),	Model		((class		in
	96				auxi.	tools.mat	erialphysic	calproperti	es.core),
mass_fr	ractions()	(in	module		103				
	auxi.tools.chem	istry.stoichio	metry),	molar_i	mass ((auxi.tools	s.chemistr	y.thermoch	emistry.Compou
	96		1 1		attrib	oute), 99			_
masses(module	molar_r	mass()	(in	mod	ule
	auxi.tools.chem	istry.stoichioi	metry),		auxi.	tools.chei	mistry.stoi	chiometry)),
	96				96				

```
molar_mass()
                                                                                                  module Phase
                                                                                                                                                                                    (class
                                                                (in
                                                                                                                                                                                                                                            in
                                                                                                                                                auxi.tools.chemistry.thermochemistry),
                     auxi.tools.chemistry.thermochemistry),
month (auxi.core.time.TimePeriod attribute), plasma (auxi.tools.materialphysicalproperties.core.StateOfMaterialphysicalproperties.core.StateOfMaterialphysicalproperties.core.StateOfMaterialphysicalproperties.core.StateOfMaterialphysicalproperties.core.StateOfMaterialphysicalproperties.core.StateOfMaterialphysicalproperties.core.StateOfMaterialphysicalproperties.core.StateOfMaterialphysicalproperties.core.StateOfMaterialphysicalproperties.core.StateOfMaterialphysicalproperties.core.StateOfMaterialphysicalphysicalproperties.core.StateOfMaterialphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalph
                                                                                                                                                attribute), 104
mstate (auxi.modelling.process.materials.core.Material(auxi.tools.materialphysicalproperties.polynomial.Poly
                                                                                                                                                method), 109
                     attribute), 78
mu() (auxi.modelling.process.materials.core.Material (auxi.core.reporting.ReportFormat at-
                     method), 78
                                                                                                                                                tribute), 51
                                                                                                                           PolynomialModelT
                                                                                                                                                                                                     (class
                                                                                                                                                                                                                                            in
N
                                                                                                                                                auxi.tools.materialphysicalproperties.polynomial),
N_A (in module auxi.tools.physicalconstants),
                                                                                                                           PolynomialModelTTester
                                                                                                                                                                                                             (class
                     (auxi.core.objects.NamedObject
name
                                                                                                                                                auxi.tools.materialphysicalproperties.polynomial_tes
                     tribute), 50
name\ (auxi.modelling.business.models. Time Bas \textbf{pd} Model
                                                                                                                                                                              (in
                                                                                                                                                                                                                              module
                     attribute), 58
                                                                                                                                                auxi.tools.transportphenomena.dimensionlessquantit
name (auxi.modelling.business.structure.Activity
                     attribute), 59
                                                                                                                           prepare_to_run()
name (auxi.modelling.business.structure.Component
                                                                                                                                                (auxi.modelling.business.basic.BasicLoanActivity
                     attribute), 60
                                                                                                                                                method), 55
name (auxi.modelling.business.structure.Entity prepare_to_run()
                                                                                                                                                (auxi.modelling.business.models. Time Based Model\\
                     attribute), 61
name (auxi.modelling.financial.des.GeneralLedgerAccountethod), 58
                     attribute), 67
                                                                                                                           prepare_to_run()
name\ (auxi.modelling.process.materials.thermo. \\Material (auxi.modelling.business.structure.Activity) \\
                     attribute), 90
                                                                                                                                                method), 59
name (auxi.tools.chemistry.thermochemistry.Phaseepare_to_run()
                     attribute), 101
                                                                                                                                                (auxi.modelling.business.structure.Component
NamedObject (class in auxi.core.objects), 50
                                                                                                                                                method), 60
NamedObjectUnitTester\\
                                                                                (class
                                                                                                                in
                                                                                                                         prepare_to_run()
                     auxi.core.objects_test), 50
                                                                                                                                                (auxi.modelling.business.structure.Entity
nu() (auxi.modelling.process.materials.core.Material
                                                                                                                                                method), 61
                     method), 78
                                                                                                                           printout (auxi.core.reporting.ReportFormat at-
Nu()
                                                                                                  module
                                                                                                                                                tribute), 51
                     auxi.tools.transport phenomena.dimensi \textbf{PseMquantiPas} kage Unit Tester
                                                                                                                                                                                                                    (class
                                                                                                                                                                                                                                            in
                                                                                                                                                auxi.modelling.process.materials.psd_test),
                      112
                                                                                                                           PsdMaterialUnitTester
                                                                                                                                                                                                         (class
                                                                                                                                                                                                                                            in
Object (class in auxi.core.objects), 50
                                                                                                                                                auxi.modelling.process.materials.psd_test),
ObjectUnitTester
                                                                       (class
                                                                                                                in
                     auxi.core.objects_test), 50
                                                                                                                           R
P
                                                                                                                           R (in module auxi.tools.physicalconstants),
P (auxi.modelling.process.materials.thermo.MaterialPackage
                     attribute), 90
                                                                                                                           Ra()
                                                                                                                                                                               (in
                                                                                                                                                                                                                             module
P (auxi.tools.material physical properties.ide algas. Rho T\\ auxi.tools.transport phenomena. dimensionless quantitation and the state of the state
                     attribute), 106
                                                                                                                                                 113
```

```
raw_assays (auxi.modelling.process.materials.thermo.Matexialools.materialphysicalproperties.idealgas),
        attribute), 90
                                                       106
                                     module RhoTPxTester
Re()
                                                                         (class
                   (in
        auxi.tools.transportphenomena.dimensionlessquantititsols.materialphysicalproperties.idealgas_test),
                                                       107
read() (auxi.core.objects.Object method), 50
                                               RhoTTester
                                                                        (class
remove_account()
                                                       auxi.tools.materialphysicalproperties.idealgas_test),
        (auxi.modelling.financial.des.GeneralLedgerAcchant
                                               RiboudViscosityTx
        method), 67
                                                                           (class
remove_component()
                                                       auxi.tools.materialphysicalproperties.slags),
        (auxi.modelling.business.structure.Component 110
                                               RiboudViscosityTy
        method), 60
                                                                           (class
remove_component()
                                                       auxi.tools.materialphysicalproperties.slags),
        (auxi.modelling.business.structure.Entity
                                                       110
        method), 61
                                               run() (auxi.modelling.business.basic.BasicActivity
remove_entity()
                                                       method), 54
        (auxi.modelling.business.models.TimeBased)/hockilmodelling.business.basic.BasicLoanActivity
                                                       method), 56
        method), 58
render() (auxi.core.reporting.Report method), run() (auxi.modelling.business.models.TimeBasedModel
                                                       method), 58
                                               run() (auxi.modelling.business.structure.Component
Report (class in auxi.core.reporting), 51
Report
                                                       method), 60
                      (class
                                           in
                                               run() (auxi.modelling.business.structure.Entity
        auxi.modelling.financial.reporting),
                                                       method), 62
report() \ (auxi.modelling.financial.des. General Ledger Structure
        method), 67
ReportFormat (class in auxi.core.reporting),
                                               S() (auxi.tools.chemistry.thermochemistry.Compound
                                                       method), 98
        51
                                               S() (auxi.tools.chemistry.thermochemistry.CpRecord
ReportingUnitTester
                             (class
                                                       method), 99
        auxi.core.reporting_test), 51
                                           in S() (auxi.tools.chemistry.thermochemistry.Phase
                        (class
ReportType
                                                       method), 101
        auxi.modelling.financial.reporting),
                                                                  (in
                                                                                     module
                                               S()
reset() (auxi.core.time.Clock method), 52
                                                       auxi.tools.chemistry.thermochemistry),
revenue (auxi.modelling.financial.des.AccountType
        attribute), 64
                                               second (auxi.core.time.TimePeriod attribute),
rho() (auxi.modelling.process.materials.core.Material
                                                       52
        method), 78
                                               set_parent_path()
                                                       (auxi.modelling.business.structure.Activity
RhoT
                     (class
                                           in
        auxi.tools.materialphysicalproperties.idealgas), method), 59
                                               set_parent_path()
                                                       (auxi.modelling.business.structure.Component
RhoTP
                      (class
                                           in
        auxi.tools.materialphysicalproperties.idealgas), method), 61
                                               set_parent_path()
        106
                                                       (auxi.modelling.business.structure.Entity
RhoTPTester
                         (class
                                           in
        auxi.tools.materialphysicalproperties.idealgas_text)thod), 62
                                               set_parent_path()
        107
                                                       (auxi.modelling.financial.des.GeneralLedgerAccount
RhoTPx
                       (class
                                           in
```

	method), 67		method), 92		
setUp()	(auxi.core.objects_test.NamedObjectUr	nistelfde.stpe(n)	(auxi.modelling.proc	ess.materials.	thermo_test.Therm
	method), 50		method), 93		
setUp()	(auxi.core.objects_test.ObjectUnitTeste	rSh()	(in	moe	dule
	method), 51		auxi.tools.transportp	henomena.dii	mensionlessquantit
setUp()	(auxi.core.reporting_test.ReportingUnit	Tester	114		
	method), 51	SlurryM	IaterialPackageUnitT	ester (class	in
setUp()	(auxi.core.time_test.ClockUnitTester		auxi.modelling.proce	ess.materials.	slurry_test),
	method), 53		87		
setUp()	(auxi.modelling.business.basic_test.Bas	si S Auctivi M	latenia Nestie Tester	(class	in
	method), 56		auxi.modelling.proce	ess.materials.	slurry_test),
setUp()	(auxi.modelling.business.basic_test.Bas	sicLoanA	&fivityUnitTester		
	method), 56	solid (au	ixi.tools.materialphys	sicalpropertie	s.core.StateOfMatt
setUp()	(auxi.modelling.business.models_test.T	imeBase	dMröblede)nitTester		
_	method), 58	solid_ga	as (auxi.tools.material	lphysicalprop	erties.core.StateOf
setUp()	(auxi.modelling.business.structure_test.	Activity	Lattirti Bentte)r, 104		
_	method), 62	solid_lic	quid (auxi.tools.mater	rialphysicalpr	operties.core.State
setUp()	(auxi.modelling.business.structure_test.	Compon	entibita)ester4		
	method), 62		quid_gas		
setUp()	(auxi.modelling.business.structure_test.	EntityUr	n taTiesiteo rols.materialp	hysicalproper	rties.core.StateOfM
. "	method), 63	•	attribute), 104		
setUp()	(auxi.modelling.financial.des_test.Gene	rsdLid <u>d</u> bio	quAdcguntblaithester		
• "	method), 69		(auxi.tools.materialp	hysicalproper	rties.core.StateOfM
setUp()	(auxi.modelling.financial.des_test.Gene	ralLedge	eaStribcttæne Worlt Tester		
- "	method), 69	_	quid_plasma		
setUp()	(auxi.modelling.financial.des_test.Gene	ralLedge	c (Mixit:Tools rmaterialp	hysicalproper	rties.core.StateOfM
- "	method), 69		attribute), 104		
setUp()	(auxi.modelling.financial.des_test.Trans	s actid<u>n</u> Te	amplat@Unitfbedtemat	erialphysicalp	properties.core.Stat
_	method), 70	-	attribute), 104		-
setUp()	(auxi.modelling.financial.des_test.Trans	sacrifo (dul	wittlesterchemistry.the	ermochemistr	y.Phase
_	method), 70		attribute), 101		
setUp()	(auxi.modelling.financial.reporting_test	.StaneOd	MedterStructureUnid	Tester	in
- "	method), 72		auxi.tools.materialph	nysicalpropert	ties.core),
setUp()	(auxi.modelling.financial.reporting_test	Transac	tloaListUnitTester		
- "	method), 72	StoichF	unctionTester	(class	in
setUp()	(auxi.modelling.process.materials.chem	_test.Ch	enn Maoedia Bhacking ey.	katidiEeritean etry	y_test),
- "	method), 76		97	_	
setUp()	(auxi.modelling.process.materials.chem	s test:k ib	eneMateriaffibietileste	r (in mo	dule
. "	method), 77		auxi.tools.chemistry.		y),
setUp()	(auxi.modelling.process.materials.psd_	test.PsdN	Pat erialPackageUnitT	ester -	
1 ∨	method), 82		metry_coefficients()		dule
setUp()	(auxi.modelling.process.materials.psd_i	test.PsdN	kateritabblnichestristry.	stoichiometry	y),
	method), 83		96	·	
	(auxi.modelling.process.materials.slurry	ys tucist gSl(muxMateriedPackingel	Lepio Felsoe mat	at-
1 🗸	method), 87		tribute), 51	•	
setUp()	(auxi.modelling.process.materials.slurry	y <u>s</u> venb8lı	* *	ethermochemi	istry.Phase
1 🗸	method), 87	•	attribute), 101		•
setUp()	(auxi.modelling.process.materials.thern	no_test.T	* *	geUnitTester	

T	testst	r() (auxi.core.objects_test.NamedObjectUnitTester
T (auxi.modelling.process.materials.thermo.Ma		
attribute) 90	test_ado	d_activity_invalid_account_name()
tearDown() (auxi.core.objects_test.NamedObje	ctUnitTe	sauxi.modelling.business.structure_test.ComponentU
method), 50		method), 62
tearDown() (auxi.core.objects_test.ObjectUnit7	etast_ado	d_activity_valid_account_name()
method), 51		$(auxi.modelling.business.structure_test.Component U$
tearDown() (auxi.core.reporting_test.Reporting	UnitTest	emethod), 62
method), 52	test_ado	1_assay()
teststr() (auxi.core.objects_test.ObjectUn	itTester	(auxi.modelling.process.materials.chem_test.ChemM
method), 51		method), 77
teststr() (auxi.modelling.process.material	st ent rado	lese Salary Material Package Unit Tester
method), 87		(auxi.modelling.process.materials.psd_test.PsdMater
teststr() (auxi.modelling.process.material	s.thermo	_MestralermoMaterialPackageUnitTester d_assay()
method), 92	iesi_au	_assay() (auxi.modelling.process.materials.slurry_test.SlurryN
testgenerate_table_()		
(auxi.modelling.financial.reporting_tes	test add	l ledgerstructureUnitTester l_assay()
method), 72 testgenerate_table_()	test_aa	(auxi.modelling.process.materials.thermo_test.Therm
(auxi.modelling.financial.reporting_tes	t Transac	,
method), 72	test add	d_operator_1()
test_meet_exection_criteria()	_	(auxi.modelling.process.materials.chem_test.ChemM
(auxi.modelling.business.basic_test.Ba	sic Activi	· · · · · · · · · · · · · · · · · · ·
method), 56	test_ado	d_operator_1()
test_meet_exection_criteria()		(auxi.modelling.process.materials.psd_test.PsdMater
(auxi.modelling.business.basic_test.Ba	sicLoan	AMENTAN DON'T Ester
method), 56	test_ado	d_operator_1()
testmeet_exection_criteria()		(auxi.modelling.process.materials.slurry_test.SlurryN
(auxi.modelling.business.structure_test	.Activity	mathedier ⁸⁷
method), 62	test_ado	d_operator_1()
testperform_year_end_gross_profit()		$(auxi.modelling.process.materials.thermo_test.Therm$
(auxi.modelling.business.structure_test	t.EntityU	mathed), 92
method), 63	test_ado	d_operator_2()
testperform_year_end_gross_profit_and_inc	ome_sun	n(auxi) modelling.process.materials.chem_test.ChemM
(auxi.modelling.business.structure_test	t.EntityU	mategory, 16
method), 63	test_ado	d_operator_2()
testperform_year_end_income_summary()		(auxi.modelling.process.materials.thermo_test.Therm
(auxi.modelling.business.structure_test	t.EntityU	muthed), 92
method), 63	test_ado	d_operator_3()
testperform_year_end_income_tax()		(auxi.modelling.process.materials.chem_test.ChemM
(auxi.modelling.business.structure_test	t.EntityU	mathed), 76
method), 63	test_ado	d_operator_3()
testperform_year_end_procedure()		(auxi.modelling.process.materials.psd_test.PsdMater
(auxi.modelling.business.structure_test	t.EntityU	impused, 82
method), 63	test_ado	d_operator_3()
testperform_year_end_retained_earnings()		(auxi.modelling.process.materials.slurry_test.SlurryN
(auxi.modelling.business.structure_test	t.EntityU	HITESSEL, 8/
method), 63	test_add	d_operator_3()

```
(auxi.modelling.process.materials.thermtostestoTherunoMaterialRecklasteUnitTester
                                                                                                                                         (auxi.tools.chemistry.thermochemistry_test.ThermoF
                    method), 92
test_add_operator_4()
                                                                                                                                         method), 102
                    (auxi.modelling.process.materials.thermtostestoffsteracto)MaterialPackageUnitTester
                                                                                                                                         (auxi.tools.materialphysicalproperties.core_test.Data
                    method), 92
test_amount() (auxi.tools.chemistry.stoichiometry_test.StnixthEdmctionTester
                    method), 97
                                                                                                                     test_construct()
test_amount_fractions()
                                                                                                                                         (auxi.tools.materialphysicalproperties.idealgas_test.F
                    (auxi.tools.chemistry.stoichiometry_test.StoichFmethod)Tester
                    method), 97
                                                                                                                     test_construct()
                                                                                                                                         (auxi.tools.materialphysicalproperties.polynomial_te
test_amounts()
                    (auxi.tools.chemistry.stoichiometry_test.StoichFmethod)Tester
                                                                                                                     test_constructor()
                    method), 97
test_beta() (auxi.modelling.process.materials.core_test.MaterialdTestebjects_test.NamedObjectUnitTester
                    method), 79
                                                                                                                                         method), 50
test_calculate()
                                                                                                                     test_constructor()
                    (auxi.tools.materialphysicalproperties.idealgas_teatxBetafEffenter_test.ClockUnitTester
                                                                                                                                         method), 53
                                                                                                                     test_constructor()
test_calculate()
                    (auxi.tools.materialphysicalproperties.idealgas_textxRhmddeTistgrbusiness.basic_test.BasicActivityUn
                    method), 107
                                                                                                                                         method), 56
test_calculate()
                                                                                                                     test_constructor()
                    (auxi.tools.material physical properties.ideal gas\_text. \textbf{R} \textbf{Ino} \textbf{T} \textbf{Pel} \textbf{T} \textbf{In} \textbf{S} \textbf{t} \textbf{t} \textbf{u} siness. basic\_test. BasicLoan Activities and the second 
                    method), 107
                                                                                                                                         method), 56
                                                                                                                     test_constructor()
test_calculate()
                    (auxi.tools.materialphysicalproperties.idealgas_textxRhmddledsing.business.models_test.TimeBasedMo
                    method), 108
                                                                                                                                         method), 58
test_calculate()
                                                                                                                     test_constructor()
                    (auxi.tools.materialphysicalproperties.polynomialauteistnRodellomiallMndsslTittextturre_test.ActivityUnit
                                                                                                                                         method), 62
                    method), 109
test_call() (auxi.tools.materialphysicalpropertiestedeatgast_trusttobre)aTTester
                                                                                                                                         (auxi.modelling.business.structure_test.ComponentU
test_call() (auxi.tools.materialphysicalproperties.idealgasmetstokl) of PTester
                    method), 107
                                                                                                                    test_constructor()
test_call() (auxi.tools.materialphysicalproperties.idealgas(atexi.RhbdePringebtesiness.structure_test.EntityUnitTe
                    method), 107
                                                                                                                                         method), 63
test\_call() \ (auxi.tools.material physical properties tiest \underline{algans}\underline{t} test. Drho TT ester
                    method), 108
                                                                                                                                         (auxi.modelling.financial.des_test.GeneralLedgerAco
test_call() (auxi.tools.materialphysicalproperties.polynomialhtest), PolynomialModelTTester
                                                                                                                     test_constructor()
                    method), 110
test_clone() (auxi.modelling.process.materials.chem_test(@heimMdeelling)PfiokagetU.dieTextest:GeneralLedgerStru
                                                                                                                                         method), 69
                    method), 77
test_clone() (auxi.modelling.process.materials.psdsttestalNstalNatate()alPackageUnitTester
                                                                                                                                         (auxi.modelling.financial.des_test.GeneralLedgerUn
                    method), 82
test_clone() (auxi.modelling.process.materials.slurry_testn&thood)MaterialPackageUnitTester
                                                                                                                    test_constructor()
test\_clone() \ (auxi.modelling.process.materials.thermo\_te(stulk in modelling.process.materials.thermo\_te(stulk in mo
```

method), 70

method), 92

test_constructor()	method), 69
(auxi.modelling.financial.des_te	est.Transæsti <u>o</u> creatit <u>Tester</u> ponent()
method), 70	(auxi.modelling.business.structure_test.ComponentU
test_constructor()	method), 62
(auxi.modelling.process.materia	als.chemtessterentatententatepiatententententententententententententent
method), 77	(auxi.modelling.business.structure_test.EntityUnitTe
test_constructor()	method), 63
(auxi.modelling.process.materia	als.chem <u>tessterbaten Manptyalld sat/T</u> ester
method), 77	(auxi.modelling.process.materials.chem_test.ChemM
test_constructor()	method), 77
(auxi.modelling.process.materia	als.core_test_lvhratriaeFioptor_assay()
method), 79	(auxi.modelling.process.materials.psd_test.PsdMater
test_constructor()	method), 83
(auxi.modelling.process.materia	als.psd_tesstPsdeAfteteriniPsyclasgeV(hitTester
method), 82	(auxi.modelling.process.materials.slurry_test.SlurryN
test_constructor()	method), 87
	als.psd_t ess t PsdeAtatecrinH Jy <u>ni</u> ta Ssa te(r)
method), 83	(auxi.modelling.process.materials.thermo_test.Thern
test_constructor()	method), 93
	als.slurrytetetstcEduteryeMaitemalPackageUnitTester
method), 87	(auxi.modelling.business.models_test.TimeBasedMo
test_constructor()	method), 58
	als.slurrytetetstcStateryMakerget@nitTester
method), 87	(auxi.modelling.process.materials.chem_test.ChemM
test_constructor()	method), 77
	als.thermtostestestation and the control of the con
method), 92	(auxi.modelling.process.materials.psd_test.PsdMater
test_constructor()	method), 83
	als.therm æstestælterpædklage r)alUnitTester
method), 93	(auxi.modelling.process.materials.slurry_test.SlurryN
test_convert_compound()	method), 87
(auxi.tools.chemistry.stoichiom	
method), 97	(auxi.modelling.process.materials.thermo_test.Thern
test_Cp() (auxi.tools.chemistry.thermocl	
method), 102	test_create_template()
test_cpRecordstr()	(auxi.tools.materialphysicalproperties.core_test.Data
•	emistry_test.ThemmetFodotib@Tester
method), 102	test_create_transaction()
**	lproperties.idealgas <u>xiestaRledfligTisten</u> cial.des_test.GeneralLedgerUni
method), 107	method), 69
test_create() (auxi.tools.materialphysica	
method), 108	(auxi.tools.materialphysicalproperties.core_test.Data
test_create() (auxi.tools.materialphysica	
· · ·	
method), 108	test_element_mass_fraction()
· · ·	lproperties.polyn(amaxialtotekt. FlodyniotryiashMoldidiffiEes ye <u>r</u> test.StoichFunct
method), 110 test_create_account()	method), 97 test_element_mass_fractions()
iesi creaie accolmici	iesi eiemeni mass trachonso

```
method), 97
                                                                                                                                                                                                                                                                test_get_assay_total()
test_elements()
                                                                                                                                                                                                                                                                                                            (auxi.modelling.process.materials.chem_test.ChemM
                                            (auxi.tools.chemistry.stoichiometry\_test. Stoich F\textbf{method}) Tester
                                                                                                                                                                                                                                                                test_get_assay_total()
test_extract_1()
                                                                                                                                                                                                                                                                                                             (auxi.modelling.process.materials.psd_test.PsdMater
                                            (auxi.modelling.process.materials.chem\_test.ChemeMatt); i \verb§RPackageUnitTester§ (auxi.modelling.process.materials.chem.process.materials.chem.process.materials.chem.process.materials.chem.process.materials.chem.process.materials.chem.process.materials.chem.process.materials.chem.process.materials.chem.process.materials.chem.process.materials.chem.process.materials.chem.process.materials.chem.process.materials.chem.process.materials.chem.process.materials.chem.process.materials.chem.process.materials.chem.process.materials.chem.process.materials.chem.process.materials.chem.process.materials.chem.process.materials.chem.process.materials.chem.process.materials.chem.process.materials.chem.process.materials.chem.process.materials.chem.process.materials.chem.process.materials.chem.process.materials.chem.process.materials.chem.process.materials.chem.process.materials.chem.process.materials.chem.process.materials.chem.process.materials.chem.process.materials.chem.process.materials.chem.process.materials.chem.process.materials.chem.process.materials.chem.process.materials.chem.process.materials.chem.process.materials.chem.process.materials.chem.process.ma
                                            method), 77
                                                                                                                                                                                                                                                                test_get_assay_total()
                                                                                                                                                                                                                                                                                                             (auxi.modelling.process.materials.slurry_test.SlurryN
test_extract_1()
                                            (auxi.modelling.process.materials.psd\_test.Psd\textbf{\textit{MatthiadP}} a \& \textit{\textit{R}} ageUnitTester
                                            method), 82
                                                                                                                                                                                                                                                                test_get_assay_total()
                                                                                                                                                                                                                                                                                                             (auxi.modelling.process.materials.thermo_test.Thern
test_extract_1()
                                            (auxi.modelling.process.materials.slurry\_test.Slurry\_test.Slurry\_test.Slurry\_test.Slurry\_test.Slurry\_test.Slurry\_test.Slurry\_test.Slurry\_test.Slurry\_test.Slurry\_test.Slurry\_test.Slurry\_test.Slurry\_test.Slurry\_test.Slurry\_test.Slurry\_test.Slurry\_test.Slurry\_test.Slurry\_test.Slurry\_test.Slurry\_test.Slurry\_test.Slurry\_test.Slurry\_test.Slurry\_test.Slurry\_test.Slurry\_test.Slurry\_test.Slurry\_test.Slurry\_test.Slurry\_test.Slurry\_test.Slurry\_test.Slurry\_test.Slurry\_test.Slurry\_test.Slurry\_test.Slurry\_test.Slurry\_test.Slurry\_test.Slurry\_test.Slurry\_test.Slurry\_test.Slurry\_test.Slurry\_test.Slurry\_test.Slurry\_test.Slurry\_test.Slurry\_test.Slurry\_test.Slurry\_test.Slurry\_test.Slurry\_test.Slurry\_test.Slurry\_test.Slurry\_test.Slurry\_test.Slurry\_test.Slurry\_test.Slurry\_test.Slurry\_test.Slurry\_test.Slurry\_test.Slurry\_test.Slurry\_test.Slurry\_test.Slurry\_test.Slurry\_test.Slurry\_test.Slurry\_test.Slurry\_test.Slurry\_test.Slurry\_test.Slurry\_test.Slurry\_test.Slurry\_test.Slurry\_test.Slurry\_test.Slurry\_test.Slurry\_test.Slurry\_test.Slurry\_test.Slurry\_test.Slurry\_test.Slurry\_test.Slurry\_test.Slurry\_test.Slurry\_test.Slurry\_test.Slurry\_test.Slurry\_test.Slurry\_test.Slurry\_test.Slurry\_test.Slurry\_test.Slurry\_test.Slurry\_test.Slurry\_test.Slurry\_test.Slurry\_test.Slurry\_test.Slurry\_test.Slurry\_test.Slurry\_test.Slurry\_test.Slurry\_test.Slurry\_test.Slurry\_test.Slurry\_test.Slurry\_test.Slurry\_test.Slurry\_test.Slurry\_test.Slurry\_test.Slurry\_test.Slurry\_test.Slurry\_test.Slurry\_test.Slurry\_test.Slurry\_test.Slurry\_test.Slurry\_test.Slurry\_test.Slurry\_test.Slurry\_test.Slurry\_test.Slurry\_test.Slurry\_test.Slurry\_test.Slurry\_test.Slurry\_test.Slurry\_test.Slurry\_test.Slurry\_test.Slurry\_test.Slurry\_test.Slurry\_test.Slurry\_test.Slurry\_test.Slurry\_test.Slurry\_test.Slurry\_test.Slurry\_test.Slurry\_test.Slurry\_test.Slurry\_test.Slurry\_test.Slurry\_test.Slurry\_test.Slurry\_test.Slurry\_test.Slurry\_test.Slurry\_test.Slurry\_test.Slurry\_test.Slurry\_test.Slurry\_test.Slurry\_test.Slurry\_test.Slurry\_test.Slurry\_test.Slurry\_test.Slurry\_test.Slurry\_test.Slurry\_test.Slurry\_test.Slurry
                                                                                                                                                                                                                                                               test_get_child_account()
                                            method), 87
                                                                                                                                                                                                                                                                                                           (auxi.modelling.financial.des\_test.GeneralLedgerAccepts and accepts a constant and accepts a constant and accepts a constant 
test_extract_1()
                                            (auxi.modelling.process.materials.thermo_test.TheathnodMaterialPackageUnitTester
                                            method), 92
                                                                                                                                                                                                                                                                test_get_component_exists()
                                                                                                                                                                                                                                                                                                           (auxi.modelling.business.structure\_test.ComponentU
test_extract_2()
                                            (auxi.modelling.process.materials.thermo\_test.T \verb| heathnodM| at \verb| erial Package Unit Tester| auxi.modelling.process.materials.thermo\_test.T \verb| heathnodM| at erial Package Unit Tester| auxi.modelling.thermo\_test.T between Unit Tester| auxi.modelling.thermo\_test.T bet
                                                                                                                                                                                                                                                                test_get_component_not_exists()
                                            method), 92
test_extract_3()
                                                                                                                                                                                                                                                                                                            (auxi.modelling.business.structure_test.ComponentU
                                            (auxi.modelling.process.materials.thermo_test.TheathootMaterialPackageUnitTester
                                                                                                                                                                                                                                                                test_get_compound_index()
                                            method), 92
test_G() (auxi.tools.chemistry.thermochemistry_test.The (molf-molf-molf-lifestperocess.materials.chem_test.ChemMaterials.chem_test.ChemMaterials.chem_test.ChemMaterials.chem_test.ChemMaterials.chem_test.ChemMaterials.chem_test.ChemMaterials.chem_test.ChemMaterials.chem_test.ChemMaterials.chem_test.ChemMaterials.chem_test.ChemMaterials.chem_test.ChemMaterials.chem_test.ChemMaterials.chem_test.ChemMaterials.chem_test.ChemMaterials.chem_test.ChemMaterials.chem_test.ChemMaterials.chem_test.ChemMaterials.chem_test.ChemMaterials.chem_test.ChemMaterials.chem_test.ChemMaterials.chem_test.ChemMaterials.chem_test.ChemMaterials.chem_test.ChemMaterials.chem_test.ChemMaterials.chem_test.ChemMaterials.chem_test.ChemMaterials.chem.test.ChemMaterials.chem.test.ChemMaterials.chem.test.ChemMaterials.chem.test.ChemMaterials.chem.test.ChemMaterials.chem.test.ChemMaterials.chem.test.ChemMaterials.chem.test.ChemMaterials.chem.test.ChemMaterials.chem.test.ChemMaterials.chem.test.ChemMaterials.chem.test.ChemMaterials.chem.test.ChemMaterials.chem.test.ChemMaterials.chem.test.ChemMaterials.chem.test.ChemMaterials.chem.test.ChemMaterials.chem.test.ChemMaterials.chem.test.ChemMaterials.chem.test.ChemMaterials.chem.test.ChemMaterials.chem.test.ChemMaterials.chem.test.ChemMaterials.chem.test.ChemMaterials.chem.test.ChemMaterials.chem.test.ChemMaterials.chem.test.ChemMaterials.chem.test.ChemMaterials.chem.test.ChemMaterials.chem.test.ChemMaterials.chem.test.ChemMaterials.chem.test.ChemMaterials.chem.test.ChemMaterials.chem.test.ChemMaterials.chem.test.ChemMaterials.chem.test.ChemMaterials.chem.test.ChemMaterials.chem.test.ChemMaterials.chem.test.ChemMaterials.chem.test.ChemMaterials.chem.test.ChemMaterials.chem.test.ChemMaterials.chem.test.ChemMaterials.chem.test.ChemMaterials.chem.test.ChemMaterials.chem.test.ChemMaterials.chem.test.ChemMaterials.chem.test.ChemMaterials.chem.test.ChemMaterials.chem.test.ChemMaterials.chem.test.ChemMaterials.chem.test.ChemMaterials.chem.test.Chem.test.Chem.test.Chem.test.Chem.test.Chem.test.Chem.test.Chem.te
                                                                                                                                                                                                                                                                                                            method), 77
                                            method), 102
                                                                                                                                                                                                                                                               test_get_compound_index()
test_get_account()
                                            (auxi.modelling.financial.des_test.GeneralLedge(natxii.cnurdell)hitgentoress.materials.thermo_test.Therm
                                            method), 69
                                                                                                                                                                                                                                                                                                            method), 93
                                                                                                                                                                                                                                                                test_get_compound_mass()
test_get_account_decendants()
                                            (auxi.modelling.financial.des_test.GeneralLedgeratxionardelling.financial.des_test.ChemM
                                                                                                                                                                                                                                                                                                            method), 77
                                            method), 69
                                                                                                                                                                                                                                                                test_get_compound_mass()
test_get_activity_exists()
                                            (auxi.modelling.business.structure_test.ComponentklinintTedstleing.process.materials.thermo_test.Thern
                                            method), 62
                                                                                                                                                                                                                                                                                                             method), 93
                                                                                                                                                                                                                                                                test_get_datafile_references()
test_get_activity_not_exists()
                                            (auxi.modelling.business.structure_test.ComponeauklinioTestehremistry.thermochemistry_test.ThermoF
                                                                                                                                                                                                                                                                                                             method), 102
                                            method), 62
test_get_assay()
                                                                                                                                                                                                                                                                test_get_date()
                                            (auxi.modelling.process.materials.chem_test.Chem_MaterialPelpkng_debtilTelstersUnitTester
                                            method), 77
                                                                                                                                                                                                                                                                                                             method), 50
                                                                                                                                                                                                                                                                test_get_datetime()
test_get_assay()
                                            (auxi.modelling.process.materials.psd_test.PsdMateriadDackinge_Uesit.Tektek*UnitTester
                                            method), 82
                                                                                                                                                                                                                                                                                                             method), 53
test_get_assay()
                                                                                                                                                                                                                                                                test_get_datetime_at_period_ix()
                                            (auxi.modelling.process.materials.slurry_test.Slu(auxi)laterialPacktegeUhitRestritTester
                                            method), 87
                                                                                                                                                                                                                                                                                                            method), 53
                                                                                                                                                                                                                                                                test_get_element_masses()
test_get_assay()
                                            (auxi.modelling.process.materials.thermo_test.ThermioModellinlPprkageLimitEriads.chem_test.ChemMaterials.thermo_test.ThermioModellinlPprkageLimitEriads.chem_test.ChemMaterials.thermo_test.ThermioModellinlPprkageLimitEriads.chem_test.ChemMaterials.thermo_test.ThermioModellinlPprkageLimitEriads.chem_test.ChemMaterials.thermo_test.ThermioModellinlPprkageLimitEriads.chem_test.ChemMaterials.thermioModellinlPprkageLimitEriads.chem_test.ChemMaterials.thermioModellinlPprkageLimitEriads.chem_test.ChemMaterials.thermioModellinlPprkageLimitEriads.chem_test.ChemMaterials.thermioModellinlPprkageLimitEriads.chem_test.ChemMaterials.thermioModellinlPprkageLimitEriads.chem_test.ChemMaterials.thermioModellinlPprkageLimitEriads.chem_test.ChemMaterials.thermioModellinlPprkageLimitEriads.chem_test.ChemMaterials.thermioModellinlPprkageLimitEriads.chem.test.ChemMaterials.thermioModellinlPprkageLimitEriads.chem.test.ChemMaterials.thermioModellinlPprkageLimitEriads.chem.test.ChemMaterials.thermioModellinlPprkageLimitEriads.chem.test.ChemMaterials.thermioModellinlPprkageLimitEriads.chem.test.ChemMaterials.thermioModellinlPprkageLimitEriads.chem.test.ChemMaterials.thermioModellinlPprkageLimitEriads.chem.test.ChemMaterials.thermioModellinlPprkageLimitEriads.chem.test.ChemMaterials.thermioModellinlPprkageLimitEriads.chem.test.ChemMaterials.thermioModellinlPprkageLimitEriads.chem.test.Chem.test.Chem.test.Chem.test.Chem.test.Chem.test.Chem.test.Chem.test.Chem.test.Chem.test.Chem.test.Chem.test.Chem.test.Chem.test.Chem.test.Chem.test.Chem.test.Chem.test.Chem.test.Chem.test.Chem.test.Chem.test.Chem.test.Chem.test.Chem.test.Chem.test.Chem.test.Chem.test.Chem.test.Chem.test.Chem.test.Chem.test.Chem.test.Chem.test.Chem.test.Chem.test.Chem.test.Chem.test.Chem.test.Chem.test.Chem.test.Chem.test.Chem.test.Chem.test.Chem.test.Chem.test.Chem.test.Chem.test.Chem.test.Chem.test.Chem.test.Chem.test.Chem.test.Chem.test.Chem.test.Chem.test.Chem.test.Chem.test.Chem.test.Chem.test.Chem.test.Chem.test.Chem.test.Chem.test.Chem.test.Chem.test.Chem.test.Chem.tes
                                            method), 93
                                                                                                                                                                                                                                                                                                            method), 77
```

```
test_get_element_masses()
                                                                                                                                                                                                          test_getitem_exists()
                                    (auxi.modelling.process.materials.thermo_test.Thermion of the limit of
                                   method), 93
                                                                                                                                                                                                                                             method), 63
test\_get\_H() \ (auxi.modelling.process.materials.theat\underline{moettiest} \\ \underline{\texttt{TlmentmenVistas}} \\ \underline{\texttt{e}} \\ \underline{\texttt{rialPackageUnitTester}} \\ \underline
                                   method), 92
                                                                                                                                                                                                                                             (auxi.modelling.business.structure_test.EntityUnitTe
                                                                                                                                                                                                                                             method), 63
test_get_mass()
                                   (auxi.modelling.process.materials.chemtetest HCI) (auxi.modelling.process.materials.chemtetest H
                                                                                                                                                                                                                                             method), 102
                                   method), 77
                                                                                                                                                                                                          test_invalid_characters()
test_get_mass()
                                   (auxi.modelling.process.materials.psd_test.PsdMaterialPdskdgethinitFestcichiometry_test.StoichFunct
                                   method), 82
                                                                                                                                                                                                                                             method), 97
                                                                                                                                                                                                          test_load_data_auxi()
test_get_mass()
                                  (auxi.modelling.process.materials.slurry_test.SlurayxvitatorisadPackisspeyUthittiFastchremistry_test.ThermoF
                                   method), 87
                                                                                                                                                                                                                                             method), 102
test_get_P() (auxi.modelling.process.materials.thestmonass()ThexinonMatthingProcessItinit@interthermo_test.Th
                                   method), 93
                                                                                                                                                                                                                                             method), 93
                                                                                                                                                                                                          test_mass() (auxi.tools.chemistry.stoichiometry_test.StoichFu
test_get_path()
                                  (auxi.modelling.business.structure_test.Activity thrittFod)e97
                                                                                                                                                                                                          test_mass_fractions()
                                   method), 62
test_get_path_relative_to_module()
                                                                                                                                                                                                                                             (auxi.tools.chemistry.stoichiometry_test.StoichFunct
                                   (auxi.core.helpers_test.HelpersUnitTester
                                                                                                                                                                                                                                             method), 97
                                   method), 50
                                                                                                                                                                                                          test_masses() (auxi.tools.chemistry.stoichiometry_test.Stoich
                                                                                                                                                                                                                                             method), 97
test_get_reference()
                                   (auxi.tools.chemistry.thermochemistry_test_TheteniaHunctionTester
                                   method), 102
                                                                                                                                                                                                                                             (auxi.tools.materialphysicalproperties.core_test.Data
test_get_referenced_accounts()
                                                                                                                                                                                                                                             method), 105
                                   (auxi.modelling.business.basic_test.Bastestctnotyt_nitsst)er
                                   method), 56
                                                                                                                                                                                                                                             (auxi.tools.chemistry.stoichiometry_test.StoichFunct
test_get_referenced_accounts()
                                                                                                                                                                                                                                             method), 97
                                    (auxi.modelling.business.basic_test.Basiektoanolativitysksh)itTester
                                   method), 57
                                                                                                                                                                                                                                              (auxi.tools.chemistry.thermochemistry_test.ThermoF
test_get_referenced_accounts()
                                                                                                                                                                                                                                              method), 102
                                    (auxi.modelling.business.structure_test. Aestivity I supilificate rator()
                                   method), 62
                                                                                                                                                                                                                                             (auxi.modelling.process.materials.chem_test.ChemM
test_get_size_class_index()
                                                                                                                                                                                                                                              method), 77
                                   (auxi.modelling.process.materials.psd_tesstPsolMiaptyiabbeniaTkos(ter
                                                                                                                                                                                                                                              (auxi.modelling.process.materials.psd\_test.PsdMaterials.psd\_test.PsdMaterials.psd\_test.PsdMaterials.psd\_test.PsdMaterials.psd\_test.PsdMaterials.psd\_test.PsdMaterials.psd\_test.PsdMaterials.psd\_test.PsdMaterials.psd\_test.PsdMaterials.psd\_test.PsdMaterials.psd\_test.PsdMaterials.psd\_test.PsdMaterials.psd\_test.PsdMaterials.psd\_test.PsdMaterials.psd\_test.PsdMaterials.psd\_test.PsdMaterials.psd\_test.PsdMaterials.psd\_test.PsdMaterials.psd\_test.PsdMaterials.psd\_test.PsdMaterials.psd\_test.PsdMaterials.psd\_test.PsdMaterials.psd\_test.PsdMaterials.psd\_test.PsdMaterials.psd\_test.PsdMaterials.psd\_test.PsdMaterials.psd\_test.PsdMaterials.psd\_test.PsdMaterials.psd\_test.PsdMaterials.psd\_test.PsdMaterials.psd\_test.PsdMaterials.psd\_test.PsdMaterials.psd\_test.PsdMaterials.psd\_test.PsdMaterials.psd\_test.PsdMaterials.psd\_test.PsdMaterials.psd\_test.PsdMaterials.psd\_test.PsdMaterials.psd\_test.PsdMaterials.psd\_test.PsdMaterials.psd\_test.PsdMaterials.psd\_test.PsdMaterials.psd\_test.PsdMaterials.psd\_test.Psd.Test.Psd.Test.Psd.Test.Psd.Test.Psd.Test.Psd.Test.Psd.Test.Psd.Test.Psd.Test.Psd.Test.Psd.Test.Psd.Test.Psd.Test.Psd.Test.Psd.Test.Psd.Test.Psd.Test.Psd.Test.Psd.Test.Psd.Test.Psd.Test.Psd.Test.Psd.Test.Psd.Test.Psd.Test.Psd.Test.Psd.Test.Psd.Test.Psd.Test.Psd.Test.Psd.Test.Psd.Test.Psd.Test.Psd.Test.Psd.Test.Psd.Test.Psd.Test.Psd.Test.Psd.Test.Psd.Test.Psd.Test.Psd.Test.Psd.Test.Psd.Test.Psd.Test.Psd.Test.Psd.Test.Psd.Test.Psd.Test.Psd.Test.Psd.Test.Psd.Test.Psd.Test.Psd.Test.Psd.Test.Psd.Test.Psd.Test.Psd.Test.Psd.Test.Psd.Test.Psd.Test.Psd.Test.Psd.Test.Psd.Test.Psd.Test.Psd.Test.Psd.Test.Psd.Test.Psd.Test.Psd.Test.Psd.Test.Psd.Test.Psd.Test.Psd.Test.Psd.Test.Psd.Test.Psd.Test.Psd.Test.Psd.Test.Psd.Test.Psd.Test.Psd.Test.Psd.Test.Psd.Test.Psd.Test.Psd.Test.Psd.Test.Psd.Test.Psd.Test.Psd.Test.Psd.Test.Psd.Test.Psd.Test.Psd.Test.Psd.Test.Psd.Test.Psd.Test.Psd.Test.Psd.Test.Psd.Test.Psd.Test.Psd.Test.Psd.Test.Psd.Test.Psd.Test.Psd.Test.Psd.Test.Psd.Test.Psd.Test.Psd.Test.Psd.Test.Psd.Test.Psd.Test.Psd.Test.Psd.Test.Psd.Test.Psd.Test.Ps
                                   method), 83
test_get_size_class_index()
                                                                                                                                                                                                                                              method), 82
                                   (auxi.modelling.process.materials.slurrytettsmShlttiphMatocialtSm()tTester
                                                                                                                                                                                                                                              (auxi.modelling.process.materials.slurry_test.SlurryN
                                   method), 87
test_get_size_class_mass()
                                                                                                                                                                                                                                             method), 87
                                   (auxi.modelling.process.materials.psd_tesstPsdlMiapeyiadPacktageUnitTester
                                                                                                                                                                                                                                             (auxi.modelling.process.materials.thermo_test.Thern
                                   method), 82
test_get_size_class_mass()
                                                                                                                                                                                                                                              method), 93
                                  (auxi.modelling.process.materials.slurrytestsmShrefy MateirialRackagelihlifffesteralproperties.core_test.I
                                                                                                                                                                                                                                             method), 105
test_get_T() (auxi.modelling.process.materials.thestmpatesthefferenanddaterialPackageUnitTester
```

method), 93

(auxi.tools.materialphysicalproperties.core_test.Data

method), 52

```
method), 105
                                                                                                                                          method), 69
test_parameter_symbols()
                                                                                                                      test_reset() (auxi.core.time_test.ClockUnitTester
                    (auxi.tools.materialphysicalproperties.core_test.DocthSet)Tester
                    method), 105
                                                                                                                      test_run() (auxi.modelling.business.basic_test.BasicActivityU
test_parameter_units()
                                                                                                                                          method), 56
                    (auxi.tools.materialphysicalproperties.cocst_test) (auxi.tools.materialphysicalphysicalproperties.cocst_test) (auxi.tools.materialphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalphysicalph
                    method), 105
                                                                                                                                          method), 58
test_phase___str__()
                                                                                                                      test_run() (auxi.modelling.business.structure_test.Component
                    (auxi.tools.chemistry.thermochemistry_test.ThemmetFordetion) Tester
                                                                                                                      test_run() (auxi.modelling.business.structure_test.EntityUnit]
                    method), 102
test_plot() (auxi.tools.materialphysicalproperties.polynomialhoelst, BôlynomialModelTTester
                    method), 110
                                                                                                                      test_run_first_month()
test_prepare_to_run()
                                                                                                                                          (auxi.modelling.business.basic_test.BasicLoanActivi
                    (auxi.modelling.business.models_test.TimeBasedMtbdellUfiitTester
                    method), 58
                                                                                                                      test_run_last_month()
test_prepare_to_run()
                                                                                                                                          (auxi.modelling.business.basic_test.BasicLoanActivi
                    (auxi.modelling.business.structure_test.Activity triatflood)e.57
                    method), 62
                                                                                                                      test_run_third_month()
                                                                                                                                          (auxi.modelling.business.basic\_test.BasicLoanActivity) and the property of t
test_prepare_to_run()
                    (auxi.modelling.business.structure_test.ComponentUnit)TesTer
                    method), 63
                                                                                                                      test_S() (auxi.tools.chemistry.thermochemistry_test.ThermoF
test_prepare_to_run()
                                                                                                                                          method), 102
                    (auxi.modelling.business.structure_test.Fastity&InHTesterxi.modelling.process.materials.thermo_test.T
                                                                                                                                          method), 93
test_read() (auxi.tools.materialphysicalpropertiestestolyetomianhetost.PolynomialModelTTester
                                                                                                                                          (auxi.modelling.business.models_test.TimeBasedMo
                    method), 110
test_reference()
                                                                                                                                          method), 58
                    (auxi.tools.materialphysicalproperties.corest_test_DataSetTester
                                                                                                                                          (auxi.modelling.business.structure_test.ActivityUnit
                    method), 105
                                                                                                                                          method), 62
test_remove_account()
                    (auxi.modelling.financial.des\_test.Gene \textit{test} \underline{\textbf{LextegenAmme}} unt Unit Tester
                    method), 69
                                                                                                                                          (auxi.modelling.business.structure_test.ComponentU
                                                                                                                                          method), 63
test_remove_component()
                    (auxi.modelling.business.structure_test.@smpeneratide(t)Tester
                    method), 63
                                                                                                                                          (auxi.modelling.business.structure_test.EntityUnitTe
                                                                                                                                          method), 64
test_remove_component()
                    (auxi.modelling.business.structure_test.Fastityet/niptallessterpath()
                    method), 63
                                                                                                                                          (auxi.modelling.business.structure_test.ActivityUnit
test_remove_entity()
                                                                                                                                          method), 62
                    (auxi.modelling.business.models_test.Titext_Bresteqb\redel|bath(T)ester
                    method), 58
                                                                                                                                          (auxi.modelling.business.structure_test.ComponentU
test_render_with_file()
                                                                                                                                          method), 63
                    (auxi.core.reporting_test.ReportingUnitTessttesret_parent_path()
                    method), 52
                                                                                                                                          (auxi.modelling.business.structure_test.EntityUnitTe
test_render_without_file()
                                                                                                                                          method), 64
                    (auxi.core.reporting_test.ReportingUnitTessttesret_parent_path()
```

136 Index

test_report() (auxi.modelling.financial.des_test.GeneralLedghrSt);uctureUnitTester

(auxi.modelling.financial.des_test.GeneralLedgerAcc

test_set_T() (auxi.modelling.process.materials.t	thermo_test.ThermoMaterialPackageUnitTester
method), 93	TimeBasedModelUnitTester (class in
test_stoichiometry_coefficient()	auxi.modelling.business.models_test),
(auxi.tools.chemistry.stoichiometry_tes	
method), 97	TimePeriod (class in auxi.core.time), 52
test_stoichiometry_coefficients()	Tmax (auxi.tools.chemistry.thermochemistry.CpRecord
(auxi.tools.chemistry.stoichiometry_tes	st.StoichFattucitioteTjest@0
method), 98	Tmin (auxi.tools.chemistry.thermochemistry.CpRecord
test_subtract_operator_2()	attribute), 100
(auxi.modelling.process.materials.cher	m <u>Træsts@hteom</u> MaterialPackageldsmitTester in
method), 77	auxi.modelling.financial.des), 68
test_subtract_operator_2()	transaction_list
	test.PsdMaterialRubblingsIfinitificited reporting.ReportType
method), 82	attribute), 71
test_subtract_operator_2()	transaction_list()
	ry_test.Slu nyxVInterlelPngkfagehbiatTdeste GeneralLedger
method), 87	method), 66
test_subtract_operator_3()	TransactionList (class in
	m_test.ChaunVintedallPagkingarl&inidTreptorting),
method), 77	71
test_subtract_operator_3()	TransactionListUnitTester (class in
	_test.PsdMaaterianlodelkinge/UnitrReistlereporting_test),
method), 83	72
test_subtract_operator_3()	TransactionTemplate (class in
	ry_test.Sluary.MaterialiPachageCialtTes)e68
method), 87	TransactionTemplateUnitTester (class in
test_tick() (auxi.core.time_test.ClockUnitTester	
method), 53	TransactionUnitTester (class in
test_validate_account_name_invalid()	auxi.modelling.financial.des_test), 70
	erEdete(chapexiStoroetxicaleximistFystleermochemistry.Phase
method), 69	attribute), 101
test_validate_account_name_valid()	txt (auxi.core.reporting.ReportFormat at-
(auxi.modelling.financial.des_test.Gen	eralLedgetrSbrute)ufelUnitTester
method), 69	
test_write() (auxi.tools.materialphysicalpropert	
method), 110	unknown (auxi.tools.materialphysicalproperties.core.StateO
ThermoFunctionTester (class in	attribute), 104
auxi.tools.chemistry.thermochemistry_	
102	auxi.tools.materialphysicalproperties.slags),
ThermoMaterialPackageUnitTester (class in	111
auxi.modelling.process.materials.therm	
92	auxi.tools.materialphysicalproperties.slags),
ThermoMaterialUnitTester (class in	111
auxi.modelling.process.materials.therm	notest),
93	validate_account_names()
tick() (auxi.core.time.Clock method), 52	(auxi.modelling.financial.des.GeneralLedgerStructu
TimeBasedModel (class in	method), 67
auxi.modelling.business.models),	memou), or

W

```
week (auxi.core.time.TimePeriod attribute), 53
write() (auxi.core.objects.Object method), 50
write_compound_to_auxi_file() (in module auxi.tools.chemistry.thermochemistry), 102
```

Y

year (auxi.core.time.TimePeriod attribute), 53