auxi User Manual *Release 0.3.5rc2*

Ex Mente Technologies (Pty) Ltd.

CONTENTS

1	Introduction	1
2	Getting Started 2.1 auxi Installation	
	2.3 Getting Help	4
3	Structure 3.1 Tools	
4	auxi Reference 4.1 auxi package	49 49
5	Indices and tables	99
Ру	non Module Index	101
In	ex	103

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

auxi Installation

auxi runs on both Linux and Windows.

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.

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
```

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:

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.

CHAPTER

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.

Tools

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

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.

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 <code>auxi</code>, 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']
```

```
A14C3 ['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']

Cr203 ['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.

\[
\rightarrow\]")

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/
\rightarrow K.")
```

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 $^{\circ}C$ 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.

Material Physical Properties Tools

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

Transport Phenomena Tools

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

Heat transfer Calculations

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

Modelling Frameworks

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

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		Material	
Name	Ilmenite	Name	Coal
Compound		Compound	
Al203[S1]		C[S1]	
CaO[S]		H2[G]	
Cr203[S]		02 [G]	
Fe203[S1]		N2[G]	
Fe304[S1]		S[S1]	
FeO[S]		A1203[S1]	
K20[S]		CaO[S]	
Mg0[S]		Fe203[S1]	
MnO[S]		MgO[S]	
Na20[S1]		SiO2[S1]	
P4010[S]		==========	
SiO2[S1]			
TiO2[S1]			
V2O5[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	1.7000000e-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	0.0000000e+00
K20[S]	4.00000000e-05	1.00000000e-05	5.0000000e-05
MgO[S]	1.04000000e-02	5.80000000e-03	1.09000000e-02
MnO[S]	5.40000000e-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
======================================			
Material Mame Composition Details		ReductantB	
Name Composition Details	(mass fractions)	ReductantB 1.00000000e+00	
Name Composition Details Compound	(mass fractions) ReductantA		
Name Composition Details Compound C[S1]	(mass fractions) ReductantA 8.40973866e-01	1.00000000e+00	
Name Composition Details Compound CIS1 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] N2[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]	(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	
======================================	(mass fractions) ReductantA 8.40973866e-01 1.37955186e-02 4.94339606e-02 6.09802120e-03 2.04933390e-03 1.20884160e-03	1.00000000e+00 0.000000000e+00 0.000000000e+00 0.00000000e+00 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	kg	
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]	7.00000000e-05	5.00000000e-05	3.10000000e-04
P4010[S]	1.00000000e-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
I			

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	0.00000000e+00	0.00000000e+00
FeO[S]	0.00000000e+00	0.00000000e+00	0.00000000e+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
		4.59480810e-01	5.99250982e-01
TiO2[S1]	1.14870203e+02	4.59460610e-01	J. JJZJUJUZE UI

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			
Material	Ilmenite		
Mass	1.05000000e+03	kg	
Amount	1.08373053e+01	kmol	
Pressure	1.00000000e+00	atm	
Temperature	6.61513374e+02	°C	
Enthalpy	-2.11851075e+03	kWh	
Compound Details			
Formula	Mass	Mass Fraction	Mole Fraction
Al203[S1]	1.36082733e+01	1.29602603e-02	1.23153413e-02
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
Fe304[S1]	0.00000000e+00	0.00000000e+00	0.00000000e+00
FeO[S]	1.79603251e+02	1.71050715e-01	2.30674699e-01
K20[S]	2.95367407e-02	2.81302292e-05	2.89340215e-05
Mg0[S]	8.75839623e+00	8.34132975e-03	2.00516825e-02
MnO[S]	5.34160012e+00	5.08723821e-03	6.94823498e-03
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
TiO2[S1]	4.05620643e+02	3.86305374e-01	4.68638424e-01
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
CaO[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]	0.0000000e+00	0.00000000e+00	0.00000000e+00
FeO[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
TiO2[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 auxi.modelling.process.materials 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 entangle 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.thermomodule.

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]
Na2O[S1]
N2[G]
O2[G]
P4O10[S]
S[S1]
SiO2[S1]
TiO2[S1]
V2O5[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
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
MgO[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 printed 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	0.0000000e+00
FeO[S]	2.79000000e-01	1.91000000e-01	0.00000000e+00
K20[S]	4.0000000e-05	1.00000000e-05	5.0000000e-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]
Si02[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 Amount	0.00000000e+00 0.00000000e+00	-	
Pressure	1.00000000e+00	atm	
Temperature	2.50000000e+00	°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	0.00000000e+00	0.000000000e+00
FeO[S]	0.00000000e+00	0.00000000e+00	0.000000000e+00
K20[S]	0.00000000e+00	0.00000000e+00	0.000000000e+00
Mg0[S]	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	0.000000000e+00
	0.000000000000	0.0000000000000	0.00000000000000
	0 00000000e+00	0 000000000+00	0 00000000e+00
SiO2[S1]	0.00000000e+00	0.00000000e+00	0.00000000e+00
SiO2[S1] TiO2[S1] V2O5[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
SiO2[S1] TiO2[S1] V2O5[S] =============	0.00000000e+00	0.00000000e+00	0.00000000e+00
SiO2[S1] TiO2[S1] V2O5[S] ====================================	0.00000000e+00 0.00000000e+00	0.00000000e+00	0.00000000e+00
SiO2[S1] TiO2[S1] V2O5[S] MaterialPackage Material	0.00000000e+00 0.00000000e+00	0.00000000e+00 0.00000000e+00	0.00000000e+00
SiO2[S1] TiO2[S1] V2O5[S] ====================================	0.00000000e+00 0.00000000e+00 ===========================	0.00000000e+00 0.00000000e+00	0.00000000e+00
SiO2[S1] TiO2[S1] V2O5[S] ====================================	0.00000000e+00 0.00000000e+00 	0.00000000e+00 0.00000000e+00	0.00000000e+00
SiO2[S1] TiO2[S1] V2O5[S] ====================================	0.00000000e+00 0.00000000e+00 	0.00000000e+00 0.00000000e+00 	0.00000000e+00
SiO2[S1] FiO2[S1] V2O5[S] ====================================	0.00000000e+00 0.00000000e+00	0.00000000e+00 0.00000000e+00	0.00000000e+00
SiO2[S1] TiO2[S1] V2O5[S] ====================================	0.00000000e+00 0.00000000e+00 	0.00000000e+00 0.00000000e+00	0.00000000e+00
SiO2[S1] TiO2[S1] V2O5[S] ====================================	0.00000000e+00 0.00000000e+00	0.00000000e+00 0.00000000e+00	0.00000000e+00 0.00000000e+00 ===========================
SiO2[S1] TiO2[S1] V2O5[S] ====================================	0.00000000e+00 0.00000000e+00	0.00000000e+00 0.00000000e+00	0.00000000e+00 0.00000000e+00 ===========================
SiO2[S1] TiO2[S1] V2O5[S] ====================================	0.00000000e+00 0.00000000e+00	0.00000000e+00 0.00000000e+00	0.00000000e+00 0.0000000e+00 ============================
SiO2[S1] FiO2[S1] V2O5[S] MaterialPackage Material Mass Amount Pressure Femperature Enthalpy Compound Details Formula	0.00000000e+00 0.00000000e+00	0.00000000e+00 0.0000000e+00 kg kmol atm °C kWh Mass Fraction 0.00000000e+00	0.00000000e+00 0.0000000e+00 ============================
SiO2[S1] TiO2[S1] V2O5[S] ===================================	0.00000000e+00 0.00000000e+00 Reductant 0.00000000e+00 1.00000000e+00 2.50000000e+01 0.00000000e+00	0.00000000e+00 0.0000000e+00 kg kmol atm °C kWh Mass Fraction 0.00000000e+00	0.00000000e+00 0.0000000e+00 ============================
SiO2[S1] TiO2[S1] V2O5[S] ===================================	0.00000000e+00 0.00000000e+00	0.00000000e+00 0.0000000e+00	0.00000000e+00 0.0000000e+00 ============================
SiO2[S1] TiO2[S1] V2O5[S] ====================================	0.00000000e+00 0.00000000e+00	0.00000000e+00 0.0000000e+00	0.00000000e+00 0.0000000e+00 ============================
SiO2[S1] TiO2[S1] V2O5[S] ====================================	0.00000000e+00 0.00000000e+00	0.00000000e+00 0.0000000e+00 kg kmol atm °C kWh 0.00000000e+00 0.0000000e+00 0.0000000e+00 0.0000000e+00	0.00000000e+00 0.0000000e+00 ============================
SiO2[S1] TiO2[S1] V2O5[S] ====================================	0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00 1.00000000e+00 2.50000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00	0.00000000e+00 0.0000000e+00 0.0000000e+00 kg kmol atm °C kWh Mass Fraction 0.00000000e+00 0.0000000e+00 0.0000000e+00 0.0000000e+00 0.00000000e+00 0.00000000e+00	0.00000000e+00 0.00000000e+00 ===========================

MgO[S]	0.00000000e+00	0.0000000e+00	0.00000000e+00
SiO2[S1]	0.00000000e+00	0.00000000e+00	0.00000000e+00
======== MaterialPackage 			
Material	Mix		
Mass	0.0000000e+00	kg	
Amount	0.00000000e+00	kmol	
Pressure	1.00000000e+00		
Temperature	2.50000000e+01		
Enthalpy	0.00000000e+00	kWh	
Compound Details			
Formula	Mass	Mass Fraction	Mole Fraction
Al203[S1]	0.00000000e+00	0.0000000e+00	0.00000000e+00
C[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	0.00000000e+00	0.00000000e+00
FeO[S]	0.00000000e+00	0.00000000e+00	0.00000000e+00
H2[G]	0.00000000e+00	0.0000000e+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	0.0000000e+00	0.00000000e+00
Na20[S1]	0.00000000e+00	0.0000000e+00	0.00000000e+00
N2[G]	0.00000000e+00	0.0000000e+00	0.00000000e+00
O2[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
TiO2[S1]	0.00000000e+00	0.00000000e+00	0.00000000e+00
V205[S]	0.00000000e+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
```

```
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)

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

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.

0)
```

```
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]
                0.00000000e+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	Mole Fraction
Al203[S1]	3.48725349e+00	7.74945219e-03	6.32625154e-03
CaO[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 swapping 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")
```

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
O2[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
V205[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.

o)

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
CaO[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
V2O5[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]
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 0.0000000e+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	ka	
Amount	2.64612753e+00	_	
Pressure	1.00000000e+00		
Temperature	2.50000000e+01		
Enthalpy	-5.15859089e+02		
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	1.00000000e+00	atm	
Temperature	2.50000000e+01	°C	
Enthalpy	-1.71953030e+02	kWh	
Compound Details	5		
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]	0.00000000e+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
V2O5[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	:=========		
======================================	Ilmenite		=========
Mass	1.56601732e+02	kg	
Amount	1.73267975e+00	-	
Pressure	1.00000000e+00	atm	
Temperature	2.50000000e+01	°C	
Enthalpy	-2.16620609e+02	kWh	
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	0.0000000e+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	0.0000000e+00
V205[S] ========	1.08225108e+00	6.91085003e-03	3.43419366e-03
	=======================================		========
========= MaterialPackage			
=======================================	Ilmenite		
======================================	Ilmenite 1.43398268e+02		
======================================		-	
======================================	1.43398268e+02	kmol	
======================================	1.43398268e+02 1.79549029e+00	kmol atm	
======================================	1.43398268e+02 1.79549029e+00 1.00000000e+00	kmol atm °C	
Material Mass Amount Pressure Temperature Enthalpy	1.43398268e+02 1.79549029e+00 1.00000000e+00 2.50000000e+01	kmol atm °C	
======================================	1.43398268e+02 1.79549029e+00 1.00000000e+00 2.50000000e+01	kmol atm °C	Mole Fraction
Material Mass Amount Pressure Temperature Enthalpy Compound Details Formula	1.43398268e+02 1.79549029e+00 1.00000000e+00 2.50000000e+01 -4.71191509e+02	kmol atm °C kWh	Mole Fraction 0.00000000e+00
Material Mass Amount Pressure Temperature Enthalpy Compound Details Formula Al203[S1]	1.43398268e+02 1.79549029e+00 1.00000000e+00 2.50000000e+01 -4.71191509e+02	kmol atm °C kWh Mass Fraction	
======================================	1.43398268e+02 1.79549029e+00 1.00000000e+00 2.50000000e+01 -4.71191509e+02 Mass	kmol atm °C kWh Mass Fraction 0.00000000e+00	0.00000000e+00
Material Mass Amount Pressure Temperature Enthalpy Compound Details Formula Al203[S1] Ca0[S] Cr203[S]	1.43398268e+02 1.79549029e+00 1.00000000e+00 2.50000000e+01 -4.71191509e+02 	kmol atm °C kWh Mass Fraction 0.00000000e+00 0.00000000e+00	0.00000000e+00
Material Mass Amount Pressure Temperature Enthalpy Compound Details Formula Al203[S1] Ca0[S] Cr203[S] Fe203[S1]	1.43398268e+02 1.79549029e+00 1.00000000e+00 2.50000000e+01 -4.71191509e+02 	kmol atm °C kWh Mass Fraction 0.00000000e+00 0.00000000e+00 0.00000000e+00	0.00000000e+00 0.00000000e+00 0.00000000e+00
Material Mass Amount Pressure Temperature Enthalpy	1.43398268e+02 1.79549029e+00 1.00000000e+00 2.50000000e+01 -4.71191509e+02 	kmol atm °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
Material Mass Amount Pressure Temperature Enthalpy Compound Details Formula Al203[S1] Ca0[S] Cr203[S] Fe203[S1] Fe304[S1]	1.43398268e+02 1.79549029e+00 1.00000000e+00 2.50000000e+01 -4.71191509e+02 Mass 0.000000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00	kmol atm °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

MnO[S]	0.0000000e+00	0.00000000e+00	0.00000000e+00
Na20[S1]	0.0000000e+00	0.00000000e+00	0.00000000e+00
P4010[S]	0.0000000e+00	0.00000000e+00	0.00000000e+00
SiO2[S1]	0.0000000e+00	0.00000000e+00	0.00000000e+00
TiO2[S1]	1.43398268e+02	1.00000000e+00	1.00000000e+00
V2O5[S]	0.0000000e+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
MaterialPackage			
Material	Ilmenite		
Mass	1.10000000e+02	kg	
Amount	1.37731044e+00	kmol	
Pressure	1.00000000e+00		
	2.50000000e+01	°C	
Temperature	2.3000000000000		
Temperature Enthalpy	-3.61448340e+02		
-			
Enthalpy			Mole Fraction
EnthalpyCompound Details	-3.61448340e+02	kWh	Mole Fraction 0.000000000e+00
Enthalpy Compound Details Formula	-3.61448340e+02	kWh Mass Fraction	
Enthalpy Compound Details FormulaAl203[S1]	-3.61448340e+02 Mass 0.00000000e+00	Mass Fraction 0.00000000e+00	0.00000000e+00
Enthalpy	-3.61448340e+02 Mass 0.00000000e+00 0.00000000e+00	Mass Fraction 0.00000000e+00 0.00000000e+00	0.00000000e+00
Enthalpy	-3.61448340e+02 Mass 0.000000000e+00 0.00000000e+00 0.00000000e+00	Mass Fraction 0.00000000e+00 0.00000000e+00 0.00000000e+00	0.00000000e+00 0.00000000e+00 0.00000000e+00
Enthalpy Compound Details Formula Al203[S1] CaO[S]	-3.61448340e+02 Mass 0.000000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00	Mass Fraction 0.000000000e+00 0.000000000e+00 0.00000000e+00 0.000000000e+00	0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
Enthalpy	-3.61448340e+02 Mass 0.000000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00	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 0.00000000e+00
Enthalpy	-3.61448340e+02 Mass 0.000000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00	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 0.00000000e+00 0.00000000e+00
Enthalpy	Mass 0.000000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00	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 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
Enthalpy	-3.61448340e+02	Mass Fraction 0.000000000e+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 0.000000000e+00 0.000000000e+00 0.000000000e+00 0.00000000e+00 0.00000000e+00
Enthalpy	-3.61448340e+02 Mass 0.000000000e+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	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 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
Enthalpy	Mass 0.000000000e+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	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 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
Enthalpy	Mass 0.000000000e+00 0.00000000e+00	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 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	0.00000000e+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
V205[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
A1203[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
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
V2O5[S]	1.08225108e+00	3.60750361e-03	1.68652807e-03
==========		=========	========

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

auxi package

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
class auxi.core.objects.Object
     Bases: object
     Base class for all auxi classes.
     static 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 = \langle 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.
class auxi.core.reporting_test.ReportingUnitTester (methodName='runTest')
     Bases: unittest.case.TestCase
     The unit tester for the class being tested.
     setUp()
```

```
tearDown()
test_render_with_file()
   Test whether a value is return
```

Test whether a value is returned when a report is rendered in a string format. And whether 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 whether any of the other formats throw an error.

auxi.core.time module

This module provides classes to manage time.

```
 \begin{array}{ll} \textbf{class} \; \texttt{auxi.core.time.Clock} \; (\textit{name}, & \textit{description=None}, \\ & \textit{start\_datetime=datetime.datetime}(1, \; 1, \; 1, \; 0, \; 0), \\ & \textit{timestep\_period\_duration=<TimePeriod.month:} \\ & 7>, \; \textit{timestep\_period\_count=1}) \\ & \textbf{Bases:} \; \textit{auxi.core.objects.NamedObject} \end{array}
```

Represents a clock that provides functions to manage a ticking clock based on a time period as well as retrieve the current tick's date since the start date.

```
get_datetime()

Get the clock's current datetime.
```

Returns The datetime.

```
{\tt 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.

class auxi.modelling.business.basic.BasicActivity(name,

```
dt_account,

cr_account,

amount=0,

start=datetime.datetime(1,

1, 1, 0, 0),

end=datetime.datetime(9999,

12, 31, 23, 59,

59, 999999),

interval=1,

descrip-

tion=None)
```

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_execution_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.

```
class auxi.modelling.business.basic_test.BasicLoanActivityUnitTester (methodNo
Bases: unittest.case.TestCase
```

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

```
setUp()
```

```
test__meet_execution_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.

riod_duration=<TimePeriod.year: 8>, pe-

8>, period_count=1)

Bases: auxi.core.objects.NamedObject

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.

- **ql_structure** The entity's general ledger structure.
- **description** The entity description.

Returns The created entity.

```
prepare_to_run()
    Prepare the model for execution.

remove_entity(name)
    Remove an entity from the model.

Peremeters remove The name (
```

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 time period.

```
test set name()
```

Test whether 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.

class auxi.modelling.business.structure.Activity(name,

start=datetime.datetime(1, 1, 1, 0, 0), end=datetime.datetime(9999, 12, 31, 23, 59, 59, 999999), interval=1, description=None)

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_accounts()

Retrieve the general ledger accounts referenced in this instance.

Returns The referenced accounts.

name

```
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

```
\begin{array}{c} \textbf{class} \; \texttt{auxi.modelling.business.structure.Component} \; (\textit{name}, & \textit{gl}, \\ & \textit{description=None}) \end{array}
```

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.

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 transactions.

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
class auxi.modelling.business.structure_test.ActivityUnitTester(methodName='r
     Bases: unittest.case.TestCase
    Tester for the auxi.modelling.business.structure.Activity class.
     setUp()
    test meet execution criteria()
    test constructor()
    test_get_path()
    test_get_referenced_accounts()
    test_prepare_to_run()
    test_set_name()
         Test whether the name changes when it is set and that the activity's path is updated
         correctly.
    test_set_parent_path()
class auxi.modelling.business.structure_test.ComponentUnitTester (methodName=
     Bases: unittest.case.TestCase
    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 that the component run's its activities' prepare_to_run methods.

test_get_activity_not_exists()

test_get_component_not_exists()

test_get_component_exists()

test_prepare_to_run()

```
test_remove_component()
     test run()
         Test that the component runs its activities.
     test_set_name()
         Test whether 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='run')
     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 whether 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 AccountType

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, struc-
ture, descrip-
tion=None)
```

Bases: auxi.core.objects.NamedObject

Represents the account structure of a general ledger.

Parameters

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

balance_sheet (end=datetime.datetime(9999, 12, 31, 23, 59, 59, 999999), for- $<math>mat=<ReportFormat.printout: (1,)>, output_path=None)$ 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.

Create a transaction in the general ledger.

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.

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 (
$$start=datetime.datetime(1, 1, 1, 0, 0)$$
), $end=datetime.datetime(9999, 12, 31, 23, 59, 59, 999999)$, $format=$, $component_path=$ '', $output_path=None$) 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.

class auxi.modelling.financial.des.GeneralLedgerAccount (name,

description=None,
number=None,
account_type=<AccountType.r
5>)

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,

descrip-

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_descendants(account)

Retrieves an account's descendants 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 descendants of the account.

validate_account_names (names)

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

Parameters names – The names of the accounts.

Returns The descendants of the account.

```
 \begin{array}{lll} \textbf{class} \ \textbf{auxi.modelling.financial.des.Transaction} \ (\textit{name}, & \textit{descrip-tion=None}, \\ & \textit{tx\_date=datetime.date} (1, \\ 1, & 1), \\ & \textit{dt\_account=None}, \\ & \textit{cr\_account=None}, \\ & \textit{source=None}, \\ & \textit{amount=0.0}, \\ & \textit{is\_closing\_dt\_account=False}, \\ & \textit{is\_closing\_cr\_account=False}) \end{array}
```

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 whether this is a closing debit account.
- is_closing_cr_account Specifies whether this is a closing credit account.

class auxi.modelling.financial.des.TransactionTemplate(name,

dt_account,
 cr_account,
 descrip tion=None)

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 (metho
    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(men
    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_descendants()
    test_report()
    test_validate_account_name_invalid()
    test_validate_account_name_valid()
class auxi.modelling.financial.des test.GeneralLedgerUnitTester(methodName='r
    Bases: unittest.case.TestCase
    Tester for the auxi.modelling.financial.des.GeneralLedger class.
    setUp()
    test_constructor()
    test_create_transaction()
class auxi.modelling.financial.des_test.TransactionTemplateUnitTester (methods
    Bases: unittest.case.TestCase
    Tester for the auxi.modelling.financial.des.TransactionTemplate class.
    setUp()
    test_constructor()
```

```
class auxi.modelling.financial.des_test.TransactionUnitTester(methodName='run')
     Bases: unittest.case.TestCase
     Tester for the auxi.modelling.financial.des.Transaction class.
     setUp()
     test_constructor()
auxi.modelling.financial.reporting module
This module provides classes to manage currencies.
class auxi.modelling.financial.reporting.BalanceSheet (data_source,
                                                                  end=datetime.date(9999,
                                                                  12,
                                                                       31),
                                                                  out-
                                                                  put_path=None)
     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.
class auxi.modelling.financial.reporting.GeneralLedgerStructure (data_source,
                                                                              out-
                                                                              put_path=None)
     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.
class auxi.modelling.financial.reporting.IncomeStatement (data_source,
                                                                      start = datetime.date(1,
                                                                      1, 1),
                                                                      end=datetime.date(9999,
                                                                      12,
                                                                      31),
                                                                      com-
```

Bases: auxi.modelling.financial.reporting.Report

po-

out-

nent_path='',

put_path=None)

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_()
```

Tester for the auxi.modelling.financial.reporting.TransactionList class.

```
setUp()
test__generate_table_()
```

Module contents

auxi.modelling.financial Package

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 Material Package.

```
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.

class auxi.modelling.process.materials.chem.MaterialPackage (material,

com-

pound_masses)

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.

```
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_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(meth
    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_empty_assay()
    test_get_assay_total()
    test_get_compound_index()
```

auxi.modelling.process.materials.core module

auxi.modelling.process.materials.core_test module

auxi.modelling.process.materials.psd module

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

- •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
                   0.18
600.0E-6
           0.06
210.0E-6
           0.02
                   0.50
           0.10
75.0E-6
                   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 Material Package 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.

class auxi.modelling.process.materials.psd.MaterialPackage (material,

size_class_masses)

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

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

auxi.modelling.process.materials.slurry module

test_get_size_class_index()

test_create_empty_assay()

test_create_package()

test_get_assay_total()

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

test_clone()

test constructor()

test_extract_1()

- 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	3.00	ш
Н2О	0.00	0.00	0.80	0.60	u
→ 1.00 307.2E-3	0.20	0.02	0.20	0.02	u
→ 0.00 108.6E-3	0.18	0.06	0.18	0.06	
→ 0.00 38.4E-3	0.17	0.04	0.17	0.04	
→ 0.00 13.6E-3	0.07	0.03	0.07	0.03	
→ 0.00 4.8E-3	0.13	0.03	0.13	0.03	
→ 0.00 1.7E-3	0.07	0.04	0.07	0.04	
→ 0.00 600.0E-6	0.06	0.18	0.06	0.18	
→ 0.00					
210.0E-6 → 0.00	0.02	0.50	0.02	0.50	ш
75.0E-6	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.

class auxi.modelling.process.materials.slurry.MaterialPackage(material,

solid_density, H2O_mass, size class masses)

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
```

```
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 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_get_assay()
test_get_mass()
test_get_size_class_mass()
test_multiply_operator()
test_subtract_operator_2()
test_subtract_operator_3()
```

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

auxi.modelling.process.materials.thermo_test module

Module contents

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.
class auxi.tools.chemistry.stoichiometry.Compound (group,
                                                            group=None,
                                                            phase=None)
     Bases: object
     Represents a full compound formula
         Parameters
               • group – iterable of Group/Element
               • dottedgroup – A Group if there is a .H2O part, None otherwise
               • phase – The phase if there is a [phase] part, None otherwise
     count()
     molar_mass()
class auxi.tools.chemistry.stoichiometry.CompoundVisitor
     Bases: parsimonious.nodes.NodeVisitor
     Visitor which takes parsed tree to useful groups.
     For parallels, check the grammar.
     generic_visit (node, other)
     visit_compound(_, compound)
     visit_dottedgroup (_, dottedgroup)
     visit_element (node, _)
     visit_group(_, group)
     visit_number (node, _)
     visit_phase (node, _)
     visit_subscriptedelement (_, subscriptedelement)
     visit_subscriptedgroup (_, subscriptedgroup)
class auxi.tools.chemistry.stoichiometry.Element (period,
                                                                    group,
                                                           atomic number,
                                                           symbol,
                                                                       mo-
                                                           lar_mass)
     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.

4.1. auxi package

- atomic_number Number of protons in the element's nucleus.
- symbol Element's symbol.
- molar_mass [kg/kmol] Element's standard atomic mass.

count()

Bases: object

Represent a part of a compound formula

Parameters

- group iterable of Group/Element parts
- multiplier multiplier in subscript or prefix of group
- dotted True of the group is like '.2H2O'

count()

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,

element)

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]

Update group counts with multiplier

This is for handling atom counts on groups like (OH)2

Parameters

- groups iterable of Group/Element
- multiplier the number to multiply by

```
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 auxi is kg.

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

Returns Molar mass. [kg/kmol]

```
auxi.tools.chemistry.stoichiometry.parse_compound auxi.tools.chemistry.stoichiometry.stoichiometry_coefficient (compound, el-e-
```

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.

ment)

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.

The function tester for the stoichiometry module.

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_phase()

Test whether the phase of a compound is parsed 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

auxi.tools.chemistry.thermochemistry test module

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_test module
auxi.tools.materialphysicalproperties.gases module
auxi.tools.materialphysicalproperties.idealgas module
auxi.tools.materialphysicalproperties.idealgas_test module
auxi.tools.materialphysicalproperties.idealgas_test module
auxi.tools.materialphysicalproperties.liquids module
auxi.tools.materialphysicalproperties.polynomial_module
auxi.tools.materialphysicalproperties.polynomial_test module

auxi.tools.materialphysicalproperties.slags module

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, 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

float)

```
auxi.tools.transportphenomena.dimensionlessquantities. Nu (L: float, h: float, k: float, k: float) \rightarrow float Calculate the Nusselt number.
```

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.\mathbf{Pr} (nu: float, alpha: float) \rightarrow float
```

Calculate the Prandtl number.

Parameters

- 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, t: float
```

Calculate the Grashof number.

Parameters

94

- 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)

 $\underset{\text{float}}{\rightarrow}$

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.

Submodules

auxi.tests module

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
                                        93
                                  auxi.tools.physicalconstants, 96
auxi, 97
                                  auxi.tools.transportphenomena,
auxi.core, 53
auxi.core.helpers, 49
                                  auxi.tools.transportphenomena.dimensionle
auxi.core.helpers_test,49
                                       93
auxi.core.objects, 50
auxi.core.objects_test,50
auxi.core.reporting, 51
auxi.core.reporting_test,51
auxi.core.time, 52
auxi.core.time_test,53
auxi.modelling, 86
auxi.modelling.business, 64
auxi.modelling.business.basic_test,
     56
auxi.modelling.business.models_test,
auxi.modelling.business.structure_test,
auxi.modelling.financial, 72
auxi.modelling.financial.des_test,
auxi.modelling.financial.reporting_test,
auxi.modelling.process, 86
auxi.modelling.process.materials,
auxi.modelling.process.materials.chem_test,
auxi.modelling.process.materials.psd_test,
auxi.modelling.process.materials.slurry_test,
auxi.tools, 96
auxi.tools.chemistry, 92
auxi.tools.chemistry.stoichiometry_test,
     91
auxi.tools.materialphysicalproperties,
```

Α			assertAlmostEqual()
AccountType	(class	in	(auxi.tools.chemistry.stoichiometry_test.StoichFunct
auxi.model	ling.financial.des)	, 64	method), 91
Activity	(class	in	asset (auxi.modelling.financial.des.AccountType
auxi.model	ling.business.struc	cture),	attribute), 64
58			AT (in module auxi.modelling.financial.des),
ActivityUnitTester	(class	in	64
auxi.model	ling.business.struc	cture_test)	auxi (module), 97
62			aux1.core (module), 53
add_activity() (auxi	.modelling.busine	ss.structu	rentingen believes (module), 49
method), 60	0		aux1.core.helpers_test (module), 49
add_assay() (auxi.m	nodelling.process.r	materials.	chemi pare paiects (module), 50
method), 73	3		aux1.core.objects_test (module), 50
add_assay() (auxi.m	nodelling.process.r	materials. _]	pauxinger apporting (module), 51
method), 7	8		auxi.core.reporting_test (module), 51
add_assay() (auxi.m	nodelling.process.r	materials.	sienry: sometime (module), 52
method), 82	2		auxi.core.time_test (module), 53
add_incompatible_j	packages()		auxi.modelling (module), 86
		rials.chen	nauxit.cndellinaensipaskagedulerester
method), 70	5		auxi.modelling.business.basic (module), 53
		erials.che	manyimodelling business.basic_test (module),
method), 74	1		56
		erials.psd.	Muximappellingebusiness.models (module), 57
method), 79			auxi.modelling.business.models_test (mod-
amount (auxi.mode		c.BasicLo	panActivityle), 38
attribute), 5			auxi.modelling.business.structure (module),
amount()	(in	module	58
auxi.tools.c 88	chemistry.stoichior	netry),	auxi.modelling.business.structure_test (mod- ule), 62
$amount_fractions()$	(in	module	auxi.modelling.financial (module), 72
auxi.tools.c	chemistry.stoichior	netry),	auxi.modelling.financial.des (module), 64
88			auxi.modelling.financial.des_test (module),
amounts()	(in	module	69
auxi.tools.c	chemistry.stoichior	netry),	auxi.modelling.financial.reporting (module),
88			70
assert Almost Equal (auxi.modelling.financial.reporting_test (mod-
(auxi.mode method), 7'		erials.chen	n_test.ChelfMaterialUnitTester auxi.modelling.process (module), 86

auxi.modelling.process.n	naterials	(module),	C
86			c (in module auxi.tools.physicalconstants), 96
auxi.modelling.process.n ule), 72	naterials.cl	nem (mod-	cash_flow (auxi.modelling.financial.reporting.ReportType attribute), 71
auxi.modelling.process.n	naterials.cl	hem_test	ChemMaterialPackageUnitTester (class in
(module), 76			auxi.modelling.process.materials.chem_test),
auxi.modelling.process.n ule), 77	naterials.p	sd (mod-	76 ChemMaterialUnitTester (class in
auxi.modelling.process.n (module), 80	naterials.p	sd_test	auxi.modelling.process.materials.chem_test),
auxi.modelling.process.n ule), 81	naterials.sl	urry (mod-	clear() (auxi.modelling.process.materials.chem.MaterialPack method), 74
auxi.modelling.process.n (module), 85	naterials.sl	urry_test	clear() (auxi.modelling.process.materials.psd.MaterialPackag method), 79
auxi.tools (module), 96			clear() (auxi.modelling.process.materials.slurry.MaterialPack
auxi.tools.chemistry (mo	dule), 92		method), 83
auxi.tools.chemistry.stoic	chiometry	(module),	Clock (class in auxi.core.time), 52
87			ClockUnitTester (class in auxi.core.time_test),
auxi.tools.chemistry.stoic	chiometry_	_test (mod-	53
ule), 91			clone() (auxi.modelling.process.materials.chem.MaterialPack
auxi.tools.materialphysic	alproperti	es (mod-	method), 74
ule), 93	unta (madu	10) 06	clone() (auxi.modelling.process.materials.psd.MaterialPackag
auxi.tools.physicalconsta auxi.tools.transportphend			method), 79
			clone() (auxi.modelling.process.materials.slurry.MaterialPack
auxi.tools.transportpheno (module), 93	micha.um	ichsiomessq	
,			Component (class in
В			auxi.modelling.business.structure),
balance_sheet (auxi.mode	elling.fina	ncial.reporti	ng Report Type Component Unit Tester (class in
attribute), 71	_		auxi.modelling.business.structure_test),
balance_sheet()			
(auxi.modelling.	financial.d	les.GeneralL	Ledger (class in
method), 64			auxi.tools.chemistry.stoichiometry),
BalanceSheet	(class	in	87
auxi.modelling.f	inancial.re	porting),	CompoundVisitor (class in
70	. 1		auxi.tools.chemistry.stoichiometry),
BasicActivity	(class	in	87
auxi.modelling.b	usiness.ba	isic),	convert_compound() (in module
53 BasicActivityUnitTester	(cla	ss in	auxi.tools.chemistry.stoichiometry),
auxi.modelling.b	`		88
56	usiness.ua	isic_test),	count() (auxi.tools.chemistry.stoichiometry.Compound method), 87
BasicLoanActivity	(class	in	count() (auxi.tools.chemistry.stoichiometry.Element
auxi.modelling.b	ousiness.ba	ısic),	method), 88
54		1 .	count() (auxi.tools.chemistry.stoichiometry.Group
BasicLoanActivityUnitTo		class in	method), 88
auxi.modelling.b	usiness.ba	isic_test),	count_with_multiplier() (in module
56			auvi tools chemistry stoichiometry)

00	
89	auxi.tools.chemistry.stoichiometry),
create_account()	89
(auxi.modelling.financial.des.GeneralL	_
method), 66	auxi.tools.chemistry.stoichiometry),
create_component()	89
(auxi.modelling.business.structure.Con	- · ·
method), 60	auxi.modelling.business.structure),
<pre>create_component()</pre>	61
(auxi.modelling.business.structure.Enti	
method), 61	<pre>auxi.modelling.business.structure_test),</pre>
create_empty_assay()	63
(auxi.modelling.process.materials.chen	melylaiteyricaluxi.modelling.financial.des.AccountType
method), 73	attribute), 64
create_empty_assay()	expense (auxi.modelling.financial.des.AccountType
(auxi.modelling.process.materials.psd.l	
method), 78	extract() (auxi.modelling.process.materials.chem.MaterialPa
create_empty_assay()	method), 75
2 0 0	ryeMaaet(a)(auxi.modelling.process.materials.psd.MaterialPack
method), 83	method), 79
create_entity()	extract() (auxi.modelling.process.materials.slurry.MaterialPa
(auxi.modelling.business.models.Timel	· · · · · · · · · · · · · · · · · · ·
method), 57	F
arasta maskaga()	•
(auxi.modelling.process.materials.chen	m.K.in module auxi.tools.physicalconstants), 96
method), 74	G
	G
(auxi modelling process materials psd l	G (in module auxi.tools.physicalconstants), 96 Material g (in module auxi.tools.physicalconstants), 96
method), 78	
	GeneralLedger (class in
(auxi modelling process materials slurr	auxi.modelling.financial.des), 64 ry.Material GeneralLedgerAccount (class in
method), 83	· ·
	auxi.modelling.financial.des), 66
create_transaction()	GeneralLedgerAccountUnitTester (class in
(auxi.modelling.financial.des.GeneralL	auxi.modelling.financial.des_test), 69
method), 65	GeneralLedgerStructure (class in
csv (auxi.core.reporting.ReportFormat at-	auxi.modelling.financial.des), 67
tribute), 51	GeneralLedgerStructure (class in
D	auxi.modelling.financial.reporting),
_	70
day (auxi.core.time.TimePeriod attribute), 52	GeneralLedgerStructureUnitTester (class in
E	auxi.modelling.financial.des_test), 69
	GeneralLedgerStructureUnitTester (class in
Element (class in	auxi.modelling.financial.reporting_test),
auxi.tools.chemistry.stoichiometry),	72
87	GeneralLedgerUnitTester (class in
element_mass_fraction() (in module	auxi.modelling.financial.des_test), 69
auxi.tools.chemistry.stoichiometry),	generic_visit()
89	(auxi.tools.chemistry.stoichiometry.CompoundVisite
element_mass_fractions() (in module	method) 87

get_account() (auxi.modelling.financial.des.Gengeatlledelgensitrmetsredictionary()

```
(auxi.modelling.process.materials.chem.MaterialPac
        method), 67
get_account_descendants()
                                                       method), 75
        (auxi.modelling.financial.des.GeneralLegerestementremasses()
        method), 67
                                                       (auxi.modelling.process.materials.chem.MaterialPac
get_activity() (auxi.modelling.business.structure.Compomenthod), 75
        method), 60
                                               get_mass() (auxi.modelling.process.materials.chem.MaterialI
get_assay() (auxi.modelling.process.materials.chem.MateriatRatkate
                                               get_mass() (auxi.modelling.process.materials.psd.MaterialPa
        method), 75
get_assay() (auxi.modelling.process.materials.psd.MaterialPlackl)g80
                                               get_mass() (auxi.modelling.process.materials.slurry.Materiall
        method), 80
get_assay() (auxi.modelling.process.materials.slurry.Materialbabkage
                                               get_mass_fraction_solids()
        method), 84
                                                       (auxi.modelling.process.materials.slurry.MaterialPac
get_assay_total()
        (auxi.modelling.process.materials.chem.Materiahethod), 84
                                               get_path_relative_to_module()
        method), 74
                                                                               (in module
                                                       auxi.core.helpers), 49
get_assay_total()
        (auxi.modelling.process.materials.psd.Matteniaferenced_accounts()
        method), 79
                                                       (auxi.modelling.business.basic.BasicActivity
get_assay_total()
                                                       method), 54
        (auxi.modelling.process.materials.slurrygMaterialenced_accounts()
                                                       (auxi.modelling.business.basic.BasicLoanActivity
        method), 83
get_child_account()
                                                       method), 55
        (auxi.modelling.financial.des.GeneralLeglegerAfterennetd_accounts()
                                                       (auxi.modelling.business.structure.Activity
        method), 66
get_component()
                                                       method), 59
        (auxi.modelling.business.structure.Comgon_enize_class_index()
        method), 60
                                                       (auxi.modelling.process.materials.psd.Material
get_compound_index()
                                                       method), 79
        (auxi.modelling.process.materials.chemgetateiziel_class_index()
        method), 74
                                                       (auxi.modelling.process.materials.slurry.Material
get_compound_mass()
                                                       method), 83
        (auxi.modelling.process.materials.chemgetateizialPhaskagnass()
                                                       (auxi.modelling.process.materials.psd.Material Packa) \\
        method), 75
get_compound_mass_fraction()
                                                       method), 80
        (auxi.modelling.process.materials.chemgetateizialPhaskagnass()
                                                       (auxi.modelling.process.materials.slurry.MaterialPac
        method), 75
get_date() (in module auxi.core.helpers), 49
                                                       method), 84
                       (auxi.core.time.Clock get_size_class_mass_fraction()
get_datetime()
                                                       (auxi.modelling.process.materials.psd.MaterialPacka
        method), 52
get_datetime_at_period_ix()
                                                       method), 80
        (auxi.core.time.Clock
                                    method), get_size_class_mass_fraction()
                                                       (auxi.modelling.process.materials.slurry.MaterialPac
get_density() (auxi.modelling.process.materials.slurry.Materhal?ackage
        method), 84
                                               get_solid_mass()
                                                       (auxi.modelling.process.materials.slurry.MaterialPac
get_element_mass()
        (auxi.modelling.process.materials.chem.Material Pattleage 85
```

106 Index

get_volume() (auxi.modelling.process.materials.slurry.Materi

method), 75

	method), 85	mass_fra	ctions()	(in	module
get_vol	ume_fraction_solids()			emistry.stoichi	ometry),
	(auxi.modelling.process.materials.slur	•	Package	-	
~ ^	method), 85	masses()		(in	module
Gr()	(in module			emistry.stoichi	ometry),
	auxi.tools.transportphenomena.dimens		notities),	(alaga	in
Group	(class in	Material	uvi modellin	(class ig.process.mat	in erials chem)
Group	auxi.tools.chemistry.stoichiometry),		72	ig.process.mat	eriais.enem),
	88	Material	_	(class	in
		8	uxi.modellin	g.process.mat	
Н			77		•
	odule auxi.tools.physicalconstants), 96	Material		(class	in
h_bar	(in module			g.process.mat	erials.slurry),
TT 1	auxi.tools.physicalconstants), 96		31		
Helpers		Materiall	_	(class	in
hour (a	auxi.core.helpers_test), 49 uxi.core.time.TimePeriod attribute), 52		iuxi.modellin 74	g.process.mat	erials.chem),
	uxi.core.time. Timer eriod attirodic), 32	Materiall		(class	in
l			•	ig.process.mat	
income	_statement	7	79	1 5 .process.mat	onais.psa),
	(auxi.modelling.financial.reporting.Re	pontatepian	Package	(class	in
	attribute), 71		_	g.process.mat	erials.slurry),
income	_statement()		33		
	(auxi.modelling.financial.des.Generall method), 65	enlatplotli- a	b (auxi.core attribute), 51	.reporting.Rep	oortFormat
Income	Statement (class in auxi.modelling.financial.reporting),	milliseco	nd (auxi.co	re.time.TimeP	Period at-
	70		ribute), 52	e.TimePeriod	attributa)
interest	_rate (auxi.modelling.business.basic.Ba	ninute (a sicLoanA	auxi.core.tiiii	e. Hiller ellou	attribute),
	attribute), 55	-	, =	ols chemistry	stoichiometry.Compound
K		r	method), 87	-	-
k B (in	module auxi.tools.physicalconstants),	molar_m		(in	module
_ `	96		10018.CHE 9()	emistry.stoichi	omeny),
I				e.TimePeriod	attribute),
Lator ((auxi care reporting DepartFormet et	5	52		
	(auxi.core.reporting.ReportFormat attribute), 51	N			
liability	(auxi.modelling.financial.des.Account attribute), 64	TypeA (in i	module auxi.	tools.physical	constants),
	attifoate), or)6 : dall: ~	h	ala TimaDagadMadal
M			xi.modelling attribute), 58	.ousiness.mod	els.TimeBasedModel
m_u (ir	n module auxi.tools.physicalconstants),			.business.struc	cture Activity
	96		attribute), 59		7.01.0.1 10.11 1 1 ty
mass()	(in module		, ,	.business.struc	cture.Component
	auxi.tools.chemistry.stoichiometry),		nttribute), 60		<u>.</u>
	90	name (au	xi.modelling	.business.struc	cture.Entity

at	tribute), 61		R			
	i.modelling.financial.des.Gen	eralLed	geracco	Hattle auxi tools phy	sicalconstants)	96
at	ttribute), 66	·	Ra()	(in	modi	
NamedOb	oject (class in auxi.core.objects	s), 50	110()	auxi.tools.transport		
NamedOb	ojectUnitTester (class	in		94	r	1
aı	uxi.core.objects_test), 50		Re()	(in	modu	ıle
Nu()	`	nodule		auxi.tools.transport	phenomena.dim	ensionlessquantit
	uxi.tools.transportphenomena.	dimensi	ionlessqu	agetities),		
9.	3		read()	(auxi.core.objects	s.Object sta	tic
0				method), 50		
_	ass in auxi cara abjects) 50		remove	_account()		
Object (c)	ass in auxi.core.objects), 50 itTester (class	in		(auxi.modelling.fina	ancial.des.Gene	ralLedgerAccoun
•	uxi.core.objects_test), 50	111		method), 66		
	axi.core.objects_test), 50		remove_	_component()		a .
Р				(auxi.modelling.bus	siness.structure.	Component
parse_con	npound (in n	nodule		method), 60		
-	uxi.tools.chemistry.stoichiome		remove_	_component() (auxi.modelling.bus	sinoss structuro	Entity
90	•	• , ,		method), 61	siness.su ucture.	Entity
png (aux	xi.core.reporting.ReportForma	at at-	remove	_entity()		
tr	ibute), 51		Temove_	_cnuty() (auxi.modelling.bus	siness models Ti	imeBasedModel
Pr()	(in n	nodule		41 41 50		
aı	uxi.tools.transportphenomena.	dimensi	onlesson	iantities) Tauxi.core.reporting	g.Report metho	d).
94	4		()	51	9F	/,
prepare_to			Report (class in auxi.core.re	eporting), 51	
(2	nuxi.modelling.business.basic.	BasicLo	Reportiv	ity (class	-	in
	nethod), 55		_	auxi.modelling.fina	ncial.reporting)	,
prepare_to		1 00' 1	D 13.4	71		
(8	nuxi.modelling.business.model	is. I imei	report()	(auxi.modelling.fina	ancial.des.Gener	ralLedgerStructur
11.	iethod), 58			method), 67		
prepare_to	o_run() auxi.modelling.business.struct	ure Acti	ReportF	Format (class in au	xi.core.reportin	g),
	nethod), 59	urc.Acu	•	51	. •	
prepare_to			Reportin	ngUnitTester	(class	in
	nuxi.modelling.business.struct	ure.Com	n n onent-	auxi.core.reporting_		•
m	nethod), 60		Report		ass	in
prepare_to	* *			auxi.modelling.finat	nciai.reporting)	,
(a	nuxi.modelling.business.struct	ure.Enti	ty _{eset()} (/1 auxi core time Clock	k method) 52	
m	nethod), 61		revenue	(auxi.modelling.fina	ancial des Acco	untTyne
printout (a	auxi.core.reporting.ReportFort	mat at-	revenue	attribute), 64	ancial.des.7 lee6	инстурс
	ibute), 51		run() (aı	uxi.modelling.busine	ess.basic.BasicA	ctivity
PsdMateri	ialPackageUnitTester (class			method) 54		•
aı	uxi.modelling.process.materia	ls.psd_te	est) ruh() (ai	uxi.modelling.busine	ess.basic.BasicL	oanActivity
O	0		V (method), 56		J
	ialUnitTester (class	in	run() (aı	uxi.modelling.busine	ess.models.Time	eBasedModel
	uxi.modelling.process.materia	Is.psd_te	est),	method), 58		
8	1		run() (au	uxi.modelling.busine	ess.structure.Co	mponent
				method), 60		

run() (auxi.modelling.business.structure.Entity	setUp()		ncial.reportii	ng_test.General	Ledg
method), 61		method), 72			
S	setUp()	(auxi.modelling.finar method), 72	icial.reportii	ng_test.Transact	tionI
second (auxi.core.time.TimePeriod attribute), 52	setUp()	(auxi.modelling.proc method), 76	ess.material	s.chem_test.Che	emM
set_parent_path()	setUp()	(auxi.modelling.proc	ess.material	s.chem_test.Che	emM
(auxi.modelling.business.structure.Act	ivity	method), 77			
method), 59	setUp()	(auxi.modelling.proc	ess.material	s.psd_test.PsdM	1 ater
set_parent_path()		method), 80			
(auxi.modelling.business.structure.Con	n petlép ()	(auxi.modelling.proc	ess.material	s.psd_test.PsdM	1 ater
method), 60		method), 81			
set_parent_path()	setUp()	(auxi.modelling.proc	ess.material	s.slurry_test.Slu	ırryN
(auxi.modelling.business.structure.Enti	ity	method), 85			
method), 61	setUp()	(auxi.modelling.proc	ess.material	s.slurry_test.Slu	ırryN
set_parent_path()		method), 86			
(auxi.modelling.financial.des.GeneralL	eSilgerAc	ecount (in	me	odule	
method), 67		auxi.tools.transportp	henomena.d	limensionlessqu	antit
setUp() (auxi.core.objects_test.NamedObjectUp	nitTester			_	
method), 50		MaterialPackageUnitT	ester (clas	s in	
setUp() (auxi.core.objects_test.ObjectUnitTeste	•	auxi.modelling.proce			
method), 50		85		•	
setUp() (auxi.core.reporting_test.ReportingUni	t Telsteny N	Material Unit Tester	(class	in	
method), 51	•	auxi.modelling.proce	ess.materials	s.slurry test),	
setUp() (auxi.core.time_test.ClockUnitTester		85		3 = //	
method), 53	StoichF	FunctionTester	(class	in	
setUp() (auxi.modelling.business.basic_test.Base			`		
method), 56		91		<i>J</i> = *****/,	
setUp() (auxi.modelling.business.basic_test.Base	sixtboahi <i>e</i>	Auctevity Coeffictent()	(in me	odule	
method), 56		auxi.tools.chemistry.	`		
setUp() (auxi.modelling.business.models_test.T	imeBase			<i>3</i> / /	
method), 58		ometry_coefficients()	(in me	odule	
setUp() (auxi.modelling.business.structure_test			,		
method), 62	,	90		-37,	
setUp() (auxi.modelling.business.structure_test	o <mark>ama</mark> 2.	(antilinitile ster orting R	ReportForma	ıt at-	
method), 62	·oumb	tribute), 51	toporti orini		
setUp() (auxi.modelling.business.structure_test	EntityU				
method), 63					
setUp() (auxi.modelling.financial.des_test.Gene	entelar Dion	wnA) fauwit Core Tobicet	s test.Name	dObjectUnitTes	ster
method), 69	nuizeug	method), 50			
setUp() (auxi.modelling.financial.des_test.Gene	ntelar Dom	* ·	s test.Objec	tUnitTester	
method), 69	Tarbeag	method), 51	= <u>-</u> 0000.00j00		
setUp() (auxi.modelling.financial.des_test.Gene	ndGrkhete	* ·	ing test Rer	ortingUnitTeste	er
method), 69	- with the	method), 51	0		
setUp() (auxi.modelling.financial.des_test.Tran	s#estonT	* *	ects test Oh	iectUnitTester	
method), 69	Sacrium	method), 51		,00001111100001	
setUp() (auxi.modelling.financial.des_test.Tran	c tost onT	, ,	ig nrocess m	aterials slurry t	test S
method), 70	Sacamin	method), 85	.g.process.iii		
memou <i>j</i> , /0		1110u10u), UJ			

```
test__generate_table_()
                                                                                                                                                                                (auxi.modelling.process.materials.slurry_test.SlurryN
                          (auxi.modelling.financial.reporting_test.GeneralhedgedStractureUnitTester
                          method), 72
                                                                                                                                                      test_add_operator_1()
                                                                                                                                                                                (auxi.modelling.process.materials.chem\_test.Chem Materials.chem\_test.Chem Materials.chem\_test.Chem\_test.Chem Materials.chem\_test.Chem Materials.chem\_test.Chem\_test.Chem\_test.Chem\_test.Chem\_test.
test__generate_table_()
                          (auxi.modelling.financial.reporting_test.Transactionthood)JillTester
                                                                                                                                                      test_add_operator_1()
                          method), 72
test__meet_execution_criteria()
                                                                                                                                                                                (auxi.modelling.process.materials.psd_test.PsdMater
                          (auxi.modelling.business.basic_test.BasicActivity) @ thio a state of the control 
                          method), 56
                                                                                                                                                      test_add_operator_1()
test\_meet\_execution\_criteria()
                                                                                                                                                                                (auxi.modelling.process.materials.slurry_test.SlurryN
                          (auxi.modelling.business.basic_test.BasicLoanAutethtodIn&Tester
                                                                                                                                                      test_add_operator_2()
                          method), 56
test__meet_execution_criteria()
                                                                                                                                                                                (auxi.modelling.process.materials.chem_test.ChemM
                          (auxi.modelling.business.structure_test.Activityltheitflod)ef/6
                                                                                                                                                      test_add_operator_3()
                          method), 62
test__perform_year_end_gross_profit()
                                                                                                                                                                                (auxi.modelling.process.materials.chem_test.ChemM
                          (auxi.modelling.business.structure_test.EntityUnitatestell), 76
                                                                                                                                                      test_add_operator_3()
test\_perform\_year\_end\_gross\_profit\_and\_income\_sum(\texttt{naryi}) (\texttt{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.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.PsdMaterials.psd\_test.PsdMaterials.psd\_test.Psd.Test
                         (auxi.modelling.business.structure_test.EntityUnitatentel), 80
                                                                                                                                                      test_add_operator_3()
                          method), 63
                                                                                                                                                                                (auxi.modelling.process.materials.slurry_test.SlurryN
test__perform_year_end_income_summary()
                          (auxi.modelling.business.structure_test.EntityUnitatesteh), 85
                                                                                                                                                      test_amount() (auxi.tools.chemistry.stoichiometry_test.Stoich
                          method), 63
test\_perform\_year\_end\_income\_tax()
                                                                                                                                                                                method), 91
                         (auxi.modelling.business.structure\_test. \textit{\textbf{Fast}} \underline{\textbf{itylbloitile}} \underline{\textbf{sfea}} ctions()
                          method), 63
                                                                                                                                                                                (auxi.tools.chemistry.stoichiometry_test.StoichFunct
test__perform_year_end_procedure()
                                                                                                                                                                                method), 91
                          (auxi.modelling.business.structure_test.FastityIbinitiTes(ter
                          method), 63
                                                                                                                                                                                (auxi.tools.chemistry.stoichiometry_test.StoichFunct
test\_perform\_year\_end\_retained\_earnings()
                                                                                                                                                                                method), 91
                         (auxi.modelling.business.structure_test.FastityIdnit(Jastexi.modelling.process.materials.chem_test.Che
                                                                                                                                                                                method), 76
                          method), 63
test_str__() (auxi.core.objects_test.NamedObjetetsIncilTonst@r(auxi.modelling.process.materials.psd_test.PsdM
                          method), 50
                                                                                                                                                                                method), 80
test_add_activity_invalid_account_name()
                                                                                                                                                      test_clone() (auxi.modelling.process.materials.slurry_test.Slu
                         (auxi.modelling.business.structure_test.ComponentUnit)Tesfer
                          method), 62
                                                                                                                                                      test_constructor()
test\_add\_activity\_valid\_account\_name()
                                                                                                                                                                                (auxi.core.objects_test.NamedObjectUnitTester
                         (auxi.modelling.business.structure_test.ComponentUnit)Tester
                          method), 62
                                                                                                                                                      test_constructor()
test_add_assay()
                                                                                                                                                                                (auxi.core.time_test.ClockUnitTester
                          (auxi.modelling.process.materials.chem_test.ChemeMaterialUnitTester
                          method), 77
                                                                                                                                                      test_constructor()
                                                                                                                                                                                (auxi.modelling.business.basic\_test.BasicActivityUn
test_add_assay()
                         (auxi.modelling.process.materials.psd\_test.Psd\textbf{\textit{M}} \textbf{\textit{natt}} \textbf{\textit{hiad}} \textbf{\textit{D}} \textbf{\textit{J}} \textbf{\textit{rfi}} \textbf{\textit{6}} \textbf{\textit{Tester}}
                                                                                                                                                      test_constructor()
                          method), 81
test_add_assay()
                                                                                                                                                                                (auxi.modelling.business.basic_test.BasicLoanActivi
```

```
method), 56
                                                                                                 method), 91
                                                                                  test_create_account()
test_constructor()
              (auxi.modelling.business.models_test.TimeBase@MxidelbdrillTngt@mancial.des_test.GeneralLedgerAco
                                                                                                 method), 69
test_constructor()
                                                                                  test_create_component()
              (auxi.modelling.business.structure_test.Activity Unit Testedelling.business.structure_test.Component U
              method), 62
                                                                                                method), 62
                                                                                  test_create_component()
test_constructor()
              (auxi.modelling.business.structure_test.ComponeantklimintTodstleing.business.structure_test.EntityUnitTe
                                                                                                 method), 63
              method), 62
                                                                                  test_create_empty_assay()
test_constructor()
              (auxi.modelling.business.structure_test.EntityUr(ittlesiterodelling.process.materials.chem_test.ChemM
                                                                                                 method), 77
              method), 63
test_constructor()
                                                                                  test_create_empty_assay()
              (auxi.modelling.financial.des_test.GeneralLedgenalcomntlellinfepturcess.materials.psd_test.PsdMater
              method), 69
                                                                                                 method), 81
test_constructor()
                                                                                  test_create_empty_assay()
              (auxi.modelling.financial.des_test.GeneralLedgeratxiicttordelling.financial.des_test.GeneralLedgeratxiicttordelling.financial.des_test.GeneralLedgeratxiicttordelling.financial.des_test.GeneralLedgeratxiicttordelling.financial.des_test.GeneralLedgeratxiicttordelling.financial.des_test.GeneralLedgeratxiicttordelling.financial.des_test.GeneralLedgeratxiicttordelling.financial.des_test.GeneralLedgeratxiicttordelling.financial.des_test.GeneralLedgeratxiicttordelling.financial.des_test.GeneralLedgeratxiicttordelling.financial.des_test.GeneralLedgeratxiicttordelling.financial.des_test.GeneralLedgeratxiicttordelling.financial.des_test.GeneralLedgeratxiicttordelling.financial.des_test.GeneralLedgeratxiicttordelling.financial.des_test.GeneralLedgeratxiicttordelling.financial.des_test.GeneralLedgeratxiicttordelling.financial.des_test.GeneralLedgeratxiicttordelling.financial.des_test.GeneralLedgeratxiicttordelling.financial.des_test.GeneralLedgeratxiicttordelling.financial.des_test.GeneralLedgeratxiicttordelling.financial.des_test.GeneralLedgeratxiicttordelling.financial.des_test.GeneralLedgeratxiicttordelling.financial.des_test.GeneralLedgeratxiicttordelling.financial.des_test.GeneralLedgeratxiicttordelling.financial.des_test.GeneralLedgeratxiicttordelling.financial.des_test.GeneralLedgeratxiicttordelling.financial.des_test.GeneralLedgeratxiicttordelling.financial.des_test.GeneralLedgeratxiicttordelling.financial.des_test.GeneralLedgeratxiicttordelling.financial.des_test.GeneralLedgeratxiicttordelling.financial.des_test.GeneralLedgeratxiicttordelling.financial.des_test.GeneralLedgeratxiicttordelling.financial.des_test.GeneralLedgeratxiicttordelling.financial.des_test.GeneralLedgeratxiicttordelling.financial.des_test.GeneralLedgeratxiicttordelling.financial.des_test.GeneralLedgeratxiicttordelling.financial.des_test.GeneralLedgeratxiicttordelling.financial.des_test.GeneralLedgeratxiicttordelling.financial.des_test.GeneralLedgeratxiicttordelling.financial.des_test.GeneralLedgeratxiicttordelling.financial.des_test.GeneralLedg
              method), 69
                                                                                                 method), 86
test_constructor()
                                                                                  test_create_entity()
              (auxi.modelling.financial.des_test.GeneralLedgenduritTresderlling.business.models_test.TimeBasedMo
              method), 69
                                                                                                 method), 58
                                                                                  test_create_package()
test constructor()
              (auxi.modelling.financial.des_test.TransactionTetauptiateIOdetTesteprocess.materials.chem_test.ChemM
              method), 69
                                                                                                 method), 77
test_constructor()
                                                                                  test_create_package()
              (auxi.modelling.financial.des_test.TransactionUnitTeistmodelling.process.materials.psd_test.PsdMater
              method), 70
                                                                                                 method), 81
test_constructor()
                                                                                  test_create_package()
              (auxi.modelling.process.materials.chem_test.ChemMaterialBinckaproChrit.Freaterials.slurry_test.SlurryN
              method), 76
                                                                                                 method), 86
test_constructor()
                                                                                  test_create_transaction()
              (auxi.modelling.process.materials.chem_test.ChemMateridellingTestercial.des_test.GeneralLedgerUni
              method), 77
                                                                                                 method), 69
test_constructor()
                                                                                  test_element_mass_fraction()
              (auxi.modelling.process.materials.psd_test.PsdMaterialPdskdgethinitYesttichiometry_test.StoichFunct
              method), 81
                                                                                                 method), 91
test_constructor()
                                                                                  test_element_mass_fractions()
              (auxi.modelling.process.materials.psd_test.PsdMateriald/hitdTesteistry.stoichiometry_test.StoichFunct
              method), 81
                                                                                                 method), 91
test_constructor()
                                                                                  test_elements()
              (auxi.modelling.process.materials.slurry_test.SlurayxVItateliadPackissgeyLstoitEbsternetry_test.StoichFunct
                                                                                                 method), 91
              method), 85
test_constructor()
                                                                                  test_extract_1()
              (auxi.modelling.process.materials.slurry_test.SluayxVInterlallingtpestess.materials.chem_test.ChemM
              method), 86
                                                                                                 method), 76
test_convert_compound()
                                                                                  test_extract_1()
              (auxi.tools.chemistry.stoichiometry_test.StoichF(moxiomTdsleing.process.materials.psd_test.PsdMater
```

```
method), 81
                                                                                                   method), 76
test_extract_1()
                                                                                    test_get_date()
              (auxi.modelling.process.materials.slurry_test.SlurayxVlaterialdelpekagtelstnHTetpsecrsUnitTester
                                                                                                   method), 50
test_get_account()
                                                                                    test_get_datetime()
              (auxi.modelling.financial.des\_test.GeneralLedge \color= \col
              method), 69
                                                                                                   method), 53
test_get_account_descendants()
                                                                                    test_get_datetime_at_period_ix()
              (auxi.modelling.financial.des\_test.GeneralLedge (\textbf{astxii.ctoure.U} mie \underline{\textbf{Tester}} Clock Unit Tester)
              method), 69
                                                                                                   method), 53
                                                                                    test_get_element_masses()
test_get_activity_exists()
              (auxi.modelling.business.structure_test.ComponentklmitTodstleing.process.materials.chem_test.ChemM
              method), 62
                                                                                                   method), 76
test_get_activity_not_exists()
                                                                                    test_get_mass()
              (auxi.modelling.business.structure_test.ComponentklinintTodstleing.process.materials.chem_test.ChemM
                                                                                                   method), 76
              method), 62
                                                                                    test_get_mass()
test_get_assay()
              (auxi.modelling.process.materials.chem_test.ChemMateriddPinckapgoCosit.Teaterials.psd_test.PsdMater
              method), 76
                                                                                                   method), 81
test_get_assay()
                                                                                    test_get_mass()
              (auxi.modelling.process.materials.psd_test.PsdMaterialPadellingeLpritTessenaterials.slurry_test.SlurryN
                                                                                                   method), 85
              method), 81
test_get_assay()
                                                                                    test_get_path()
              (auxi.modelling.process.materials.slurry_test.SluayxVInterlalPagklagsiblesisTetsteeture_test.ActivityUnit
              method), 85
                                                                                                   method), 62
test_get_assay_total()
                                                                                    test_get_path_relative_to_module()
              (auxi.modelling.process.materials.chem_test.ChemMaterialHelpieTestest.HelpersUnitTester
              method), 77
                                                                                                   method), 50
test_get_assay_total()
                                                                                    test_get_referenced_accounts()
              (auxi.modelling.process.materials.psd_test.PsdMaterialHoditfTesgebusiness.basic_test.BasicActivityUn
              method), 81
                                                                                                   method), 56
test_get_assay_total()
                                                                                    test_get_referenced_accounts()
              (auxi.modelling.process.materials.slurry_test.SlurayxVinterlallimitTusiness.basic_test.BasicLoanActivi
              method), 86
                                                                                                   method), 56
test_get_child_account()
                                                                                    test_get_referenced_accounts()
              (auxi.modelling.financial.des_test.GeneralLedge@aluxionnotlellinffesteriness.structure_test.ActivityUnit
              method), 69
                                                                                                   method), 62
test_get_component_exists()
                                                                                    test_get_size_class_index()
              (auxi.modelling.business.structure_test.ComponeantklinintTedstleing.process.materials.psd_test.PsdMater
              method), 62
                                                                                                   method), 81
test_get_component_not_exists()
                                                                                    test_get_size_class_index()
              ______(auxi.modelling.business.structure_test.Compon@auklmint@dstleing.process.materials.slurry_test.SlurryN
              method), 62
                                                                                                   method), 86
test_get_compound_index()
                                                                                    test_get_size_class_mass()
              (auxi.modelling.process.materials.chem_test.ChemMaterialdlingTexterss.materials.psd_test.PsdMater
              method), 77
                                                                                                   method), 81
                                                                                    test_get_size_class_mass()
```

(auxi.modelling.process.materials.chem_test.ChemMaterialBinckaproChrit.Freaterials.slurry_test.SlurryN

test_get_compound_mass()

```
method), 85
                                                      method), 62
test_getitem_exists()
                                              test_remove_component()
        (auxi.modelling.business.structure_test.EntityUnitTesterodelling.business.structure_test.EntityUnitTe
        method), 63
                                                      method), 63
                                              test_remove_entity()
test_getitem_not_exists()
        (auxi.modelling.business.structure_test.EntityUr(ittTesterodelling.business.models_test.TimeBasedMo
        method), 63
                                                      method), 58
test_invalid_characters()
                                              test_render_with_file()
        (auxi.tools.chemistry.stoichiometry_test.StoichF(moxtiomoFesterporting_test.ReportingUnitTester
        method), 91
                                                      method), 52
test_mass() (auxi.tools.chemistry.stoichiometry_test_SeniderFwichiom(Testect)
        method), 91
                                                       (auxi.core.reporting_test.ReportingUnitTester
test_mass_fractions()
                                                       method), 52
        (auxi.tools.chemistry.stoichiometry_testtesteichfortottonTesterdelling.financial.des_test.GeneralLedge
                                                      method), 69
        method), 91
method), 91
                                                       method), 53
                                              test_run() (auxi.modelling.business.basic_test.BasicActivityU
test_molar_mass()
        (auxi.tools.chemistry.stoichiometry\_test.StoichF\textbf{method}) \textbf{\textit{Tester}}
        method), 92
                                              test_run() (auxi.modelling.business.models_test.TimeBasedM
test_multiply_operator()
                                                      method), 58
        (auxi.modelling.process.materials.chemtetstruchentalsingebusintesssteructure_test.Component
        method), 76
                                                      method), 63
test_multiply_operator()
                                              test_run() (auxi.modelling.business.structure_test.EntityUnit]
       (auxi.modelling.process.materials.psd\_test.Psd\textbf{\textit{MatthiadP}} a6 \textit{R} age Unit Tester
                                              test run first month()
        method), 81
test_multiply_operator()
                                                      (auxi.modelling.business.basic_test.BasicLoanActivi
        (auxi.modelling.process.materials.slurry_test.SlumetNoaterialPackageUnitTester
                                              test run last month()
        method), 85
test_phase() (auxi.tools.chemistry.stoichiometry_test.Sto(ahkiinvoidedllestabusiness.basic_test.BasicLoanActivi
        method), 92
                                                      method), 57
test_prepare_to_run()
                                              test_run_third_month()
       (auxi.modelling.business.models_test.TimeBase@ModellodeilfTegterusiness.basic_test.BasicLoanActivi
        method), 58
                                                      method), 57
test_prepare_to_run()
                                              test_set_name()
        (auxi.modelling.business.structure_test.Activity (auxi.modelling.business.models_test.TimeBasedMo
        method), 62
                                                      method), 58
                                              test_set_name()
test_prepare_to_run()
        (auxi.modelling.business.structure_test.ComponentklinitTodstleing.business.structure_test.ActivityUnit
        method), 62
                                                      method), 62
test_prepare_to_run()
                                              test_set_name()
       (auxi.modelling.business.structure_test.EntityUnittlestmodelling.business.structure_test.ComponentU
        method), 63
                                                      method), 63
test_remove_account()
                                              test_set_name()
       (auxi.modelling.financial.des_test.GeneralLedgenAuxionnotleHiffEelsteriness.structure_test.EntityUnitTe
        method), 69
                                                      method), 63
test_remove_component()
                                              test_set_parent_path()
```

(auxi.modelling.business.structure_test.Componeauklinitoelstleing.business.structure_test.ActivityUnit

method), 62	58
test_set_parent_path()	TimePeriod (class in auxi.core.time), 52
(auxi.modelling.business.structure_test	
method), 63	auxi.modelling.financial.des), 67
test_set_parent_path()	transaction_list
	EntityUn(itTexterodelling.financial.reporting.ReportType
method), 63	attribute), 71
test_set_parent_path()	transaction_list()
	eralLedge(mAuxiomntlellinfestuerncial.des.GeneralLedger
method), 69	method), 65
test_stoichiometry_coefficient()	TransactionList (class in
	st.StoichFaunctionoThesteurg.financial.reporting),
method), 92	71
test_stoichiometry_coefficients()	TransactionListUnitTester (class in
•	st.StoichFaunctionoThesteurg.financial.reporting_test),
method), 92	72
test_subtract_operator_2()	TransactionTemplate (class in
• "	n_test.ChaunWaterlellPagkfagærleinalThester68
method), 77	TransactionTemplateUnitTester (class in
test_subtract_operator_2()	auxi.modelling.financial.des_test), 69
(auxi.modelling.process.materials.psd_	, ,
method), 81	auxi.modelling.financial.des_test), 69
test_subtract_operator_2()	txt (auxi.core.reporting.ReportFormat at-
(auxi.modelling.process.materials.slurr	ry_test.SlutribyMarkerialPackageUnitTester
method), 85	V
•	
test_subtract_operator_3()	V
	•
	v n <u>velid Chem Wwerim Pack age</u> Unit Tester (auxi.modelling.financial.des.General Ledger Structur
(auxi.modelling.process.materials.chen	n <u>velid.CheraMwerinPneka</u> geUnitTester
(auxi.modelling.process.materials.chemmethod), 77	nvelid: Chem Muleria Package Unit Tester (auxi.modelling.financial.des. General Ledger Structur method), 67
(auxi.modelling.process.materials.chemmethod), 77 test_subtract_operator_3()	nvelid: Chem Muleria Package Unit Tester (auxi.modelling.financial.des. General Ledger Structur method), 67
(auxi.modelling.process.materials.chemmethod), 77 test_subtract_operator_3() (auxi.modelling.process.materials.psd_	nvelid Cheminameria Package Unit Tester (auxi.modelling.financial.des. General Ledger Structur method), 67 twist Pschripprint Package Unit Tester
(auxi.modelling.process.materials.chemmethod), 77 test_subtract_operator_3()	nvelid. Chean Mustim Package Unit Tester (auxi.modelling. financial.des. General Ledger Structur method), 67 teiri Pset Marerial Package Unit Tester (auxi.tools.chemistry.stoichiometry. Compound Visite
(auxi.modelling.process.materials.chemmethod), 77 test_subtract_operator_3()	nvelid Chera Maneria Procka ge Unit Tester (auxi.modelling.financial.des. General Ledger Structur method), 67 teixi Pset Maneria Package Unit Tester (auxi.tools.chemistry.stoichiometry. Compound Visitomethod), 87
(auxi.modelling.process.materials.chemmethod), 77 test_subtract_operator_3()	nvelst. Chean Mandrial Package Unit Tester (auxi.modelling.financial.des. General Ledger Structur method), 67 testiles composite the charge Unit Tester (auxi.tools.chemistry.stoichiometry. Compound Visite method), 87 ryvisist. Source Mandrial Package Unit Tester (auxi.tools.chemistry.stoichiometry. Compound Visite (auxi.tools.chemistry.stoichiometry. Compound Visite
(auxi.modelling.process.materials.chemmethod), 77 test_subtract_operator_3()	nvelst. Chean Mandrial Package Unit Tester (auxi.modelling.financial.des. General Ledger Structur method), 67 testiles composite that chage Unit Tester (auxi.tools.chemistry.stoichiometry. Compound Visite method), 87 ryvisist. Source Mandrial Package Unit Tester (auxi.tools.chemistry.stoichiometry. Compound Visite (auxi.tools.chemistry.stoichiometry. Compound Visite
(auxi.modelling.process.materials.chemmethod), 77 test_subtract_operator_3()	nyist shapetal package Unit Tester (auxi.modelling.financial.des.General Ledger Structur method), 67 test Pset point Cackage Unit Tester (auxi.tools.chemistry.stoichiometry.Compound Visitomethod), 87 ryist shapetal Package Unit Tester (auxi.tools.chemistry.stoichiometry.Compound Visitomethod), 87 method), 87
(auxi.modelling.process.materials.chemmethod), 77 test_subtract_operator_3()	n_velst. Chean Manetial Package Unit Tester (auxi.modelling.financial.des. General Ledger Structur method), 67 testips of Manetial Package Unit Tester (auxi.tools.chemistry.stoichiometry. Compound Visite method), 87 ryvisist. Source Manetial Package Unit Tester (auxi.tools.chemistry.stoichiometry. Compound Visite method), 87 visit_element() (auxi.tools.chemistry.stoichiometry. Compound Visite
(auxi.modelling.process.materials.chemmethod), 77 test_subtract_operator_3()	n_velst. Chean Manetial Package Unit Tester (auxi.modelling.financial.des. General Ledger Structur method), 67 testips of Manetial Package Unit Tester (auxi.tools.chemistry.stoichiometry. Compound Visite method), 87 ryvisist. Source Manetial Package Unit Tester (auxi.tools.chemistry.stoichiometry. Compound Visite method), 87 visit_element() (auxi.tools.chemistry.stoichiometry. Compound Visite
(auxi.modelling.process.materials.chemmethod), 77 test_subtract_operator_3()	n_velid.CheanMartinPackageUnitTester (auxi.modelling.financial.des.GeneralLedgerStructur method), 67 testipscharenteackageUnitTester (auxi.tools.chemistry.stoichiometry.CompoundVisite method), 87 ryvisit.SourceackageUnitTester (auxi.tools.chemistry.stoichiometry.CompoundVisite method), 87 visit_element() (auxi.tools.chemistry.stoichiometry.CompoundVisite eralLedgerSthoodrevUnitTester
(auxi.modelling.process.materials.chemmethod), 77 test_subtract_operator_3()	n_velid_Chean Manetial Package Unit Tester
(auxi.modelling.process.materials.chemmethod), 77 test_subtract_operator_3()	n_velid.CheasonetialPackageUnitTester (auxi.modelling.financial.des.GeneralLedgerStructur method), 67 testipscontrolateackageUnitTester (auxi.tools.chemistry.stoichiometry.CompoundVisite method), 87 ryvisit.compoundVisite (auxi.tools.chemistry.stoichiometry.CompoundVisite method), 87 visit_element() (auxi.tools.chemistry.stoichiometry.CompoundVisite eralLedgerstheet)reCnitTester visit_group() (auxi.tools.chemistry.stoichiometry.Compound method), 87
(auxi.modelling.process.materials.chemmethod), 77 test_subtract_operator_3()	nvelid. Chemistrus financial. des. General Ledger Structur method), 67 teisi Psannount Rackage Unit Tester (auxi. tools. chemistry. stoichiometry. Compound Visitomethod), 87 ryvisit. Sottog Mann Cal Package Unit Tester (auxi. tools. chemistry. stoichiometry. Compound Visitomethod), 87 visit_element() (auxi. tools. chemistry. stoichiometry. Compound Visitometal Ledger Structure Unit Tester visit_group() (auxi. tools. chemistry. stoichiometry. Compound method), 87 eville duration ture Unit Tester (auxi. tools. chemistry. stoichiometry. Compound Visitomethod), 87 eville duration ture Unit Tester (auxi. tools. chemistry. stoichiometry. Compound Visitomethod), 87
(auxi.modelling.process.materials.chemmethod), 77 test_subtract_operator_3()	n_velid_Chean Mandrial Package Unit Tester (auxi.modelling.financial.des.General Ledger Structur method), 67 teinipset Mandrial Package Unit Tester (auxi.tools.chemistry.stoichiometry.Compound Visite method), 87 ryv_isist_Stand Mandrial Package Unit Tester (auxi.tools.chemistry.stoichiometry.Compound Visite method), 87 visit_element() (auxi.tools.chemistry.stoichiometry.Compound Visite eral Ledger Stand Ure Unit Tester visit_group() (auxi.tools.chemistry.stoichiometry.Compound method), 87 erinie auxi.tools.chemistry.stoichiometry.Compound Visite (auxi.tools.chemistry.stoichiometry.Compound Visite compound Compound Visite (auxi.tools.chemistry.stoichiometry.Compound Visite (auxi.tools.chemistry.stoichiometry.chemistry.chemistry.stoichiometry.chemistry.chemistry.stoichiometry.chemistry.chemistry.chemistry.chemistry.chemistry.chemistry.c
(auxi.modelling.process.materials.chemmethod), 77 test_subtract_operator_3()	nythin the and the intractage Unit Tester (auxi.modelling.financial.des.GeneralLedger Structure method), 67 testiles and portal Rackage Unit Tester (auxi.tools.chemistry.stoichiometry.Compound Visite method), 87 rythis to the parametal Package Unit Tester (auxi.tools.chemistry.stoichiometry.Compound Visite method), 87 visit_element() (auxi.tools.chemistry.stoichiometry.Compound Visite eralLedgers the Ure Unit Tester visit_group() (auxi.tools.chemistry.stoichiometry.Compound method), 87 eville agurs except the Unit Tester (auxi.tools.chemistry.stoichiometry.Compound Visite method), 87 visit_phase() (auxi.tools.chemistry.stoichiometry.Compound method), 87
(auxi.modelling.process.materials.chemmethod), 77 test_subtract_operator_3()	nvelid. Chemistrate has deed unit Tester (auxi.modelling. financial.des. General Ledger Structure method), 67 test Psantage unit Tester (auxi.tools.chemistry.stoichiometry. Compound Visitamethod), 87 yvisit. Sotto yvantage unit Tester (auxi.tools.chemistry.stoichiometry. Compound Visitamethod), 87 visit_element() (auxi.tools.chemistry.stoichiometry. Compound Visitamethod), 87 visit_group() (auxi.tools.chemistry.stoichiometry. Compound method), 87 eville duer set (auxi.tools.chemistry.stoichiometry. Compound Visitamethod), 87 visit_phase() (auxi.tools.chemistry.stoichiometry. Compound Visitamethod), 87 visit_phase() (auxi.tools.chemistry.stoichiometry. Compound Visitamethod), 87
(auxi.modelling.process.materials.chemmethod), 77 test_subtract_operator_3() (auxi.modelling.process.materials.psd_method), 81 test_subtract_operator_3() (auxi.modelling.process.materials.slurr_method), 85 test_tick() (auxi.core.time_test.ClockUnitTester_method), 53 test_validate_account_name_invalid() (auxi.modelling.financial.des_test.Genemethod), 69 test_validate_account_name_valid() (auxi.modelling.financial.des_test.Genemethod), 69 tick() (auxi.core.time.Clock method), 52 TimeBasedModel (class in auxi.modelling.business.models),	nythin the and the intractage Unit Tester (auxi.modelling.financial.des.GeneralLedger Structure method), 67 testiles and portal Rackage Unit Tester (auxi.tools.chemistry.stoichiometry.Compound Visite method), 87 rythis to the parametal Package Unit Tester (auxi.tools.chemistry.stoichiometry.Compound Visite method), 87 visit_element() (auxi.tools.chemistry.stoichiometry.Compound Visite eralLedgers the Ure Unit Tester visit_group() (auxi.tools.chemistry.stoichiometry.Compound method), 87 eville agurs except the Unit Tester (auxi.tools.chemistry.stoichiometry.Compound Visite method), 87 visit_phase() (auxi.tools.chemistry.stoichiometry.Compound method), 87