ChemApp for Python Friendly Cheat Sheet

Visit documentation for more at https://python.gtt-technologies.de/doc/chemapp/

First things first – import ChemApp Packages

from **chemapp.core** import **AmountUnit, TemperatureUnit, PressureUnit, VolumeUnit, EnergyUnit**

from **chemapp.friendly** import **Units** as cau

from **chemapp.friendly** import **ThermochemicalSystem** as cats

from **chemapp.friendly** import **StreamCalculation** as casc

from **chemapp.friendly** import **EquilibriumCalculation** as caec

Get started with ThermochemicalSystem

cats.load(*path*)	Open, read and close the .cst or .dat file from *path.*
cats.get_str_phs_pcs()	Print a list of phases and phase constituents in the loaded thermochemical system to a string.
cau.set(*T=**TemperatureUnit**.C,P=**PressureUnit**.atm,A=**AmountUnit**.kg, V = **VolumeUnit**.m3, E= **EnergyUnit**.kWh*)	Set system units. Default units are presented in the example.
cau.get_T_unit().name	Get the unit of Temperature as string
cau.get()	Get units in a dictionary.
cats.get_count_pcs()	Get the total number of phase constituents in the system.
cats.get_name_pcs_in_ph(*x*)	Get the name of phase constituents in phase x.
cats.get_index_pc(*x,y*)	Get the index of the phase consituent y in the phase x.

cats.get_status_ph(*x*)
cats.set_status_ph(*x, Status.ENTERED*)
	Get/set the status of phase x. Options: ENTERED, DORMANT, ELIMINATED

Get started with EquilibriumCalculation

caec.set_IA_sc(*x, i*)	Set the system component x incoming amount to i.
caec.set_tg_formation_of_ph(*x*)	Set the target as formation of phase x.
caec.set_eq_P(*i*)	Set equilibrium pressure for the system to i.
caec.calculate_eq_T(*i*)	Search for the target by adjusting temperature, and use the provided guess *i* as starting value.
caec.calculate_eq()	Calculate the single equilibrium
caec.get_eq_A_ph(*x*)	Get the amount of phase *x*
caec.get_eq_T()	Get the temperature at equilibrium

Get started with StreamCalculation

casc.create_st(*"stream_name",T= i, P = z*)	Create a stream and set its temperature and pressure.
casc.set_IA_pc(*"stream_name"*,*x*,*y*,*i*)	Set amount of the phase constituent y in phase x to i.
casc.set_eq_T(i)	Set the equilibrium temperature for the system to i.
result=casc.calculate_eq(*print_results = True, result_object = True, stream_extensive_properties=True*)	Calculate the equilibrium and extract an equilibrium calculation's results as an object including the extensive properties.
result.ph[*x*].pc[*y*].A	Get amount of a phase constituent *y* in phase *x* from result object.
result.dH	Get the enthalpy change from result object.
casc.remove_eq_conditions_all()	Remove all equilibrium conditions

Quantity

Pressure	bar, atm, Pa, kPa, psi, torr
Volume	dm3, cm3, m3, ft3, in3
Temperature	K, C, F
Energy	J, cal, Btu, kWh
Amount	mol, gram, kg, tonne, pound

Methods

Make use of CAPy friendy packages

cats	.load(),.get_, .set_
caec	.get_, .set_, .calculate_, .remove_
casc	.get_, .set_, .calculate., remove_, .create_

Tools

Availability is based on the method cats,caec,casc

.get_	count_ , config_ , name_, status_, str_ , index_ ,y_,mm_,result_object(), results_IAs(), eq_	
.set_	status_ , eq_,IA_, tg_ , target_ ,	
.calculate_	eq(),eq_	IA(x), T(x), P(x), V(x)
.remove_	eq_conditions_	all(), IA()
	st(),sts()	
.create_	st('name',T,P)	

Entity Properties

x*,*y*,*z takes name (string) or index (integer) of the entity as an argument, and ***i*** is integer or float

System component x	sc(*x*)
All system components	scs()
Phase x	ph(*x*)
All phases	phs()
A phase constituent y in a phase x	pc(*x,y*)
All phase constituents	pcs()
Phase constituents in phase x	pcs_in_ph(*x*)
Mass fractions of the system component z in the phase constituent y in a phase x *cats.get_y_	sc_pcs(*z*), sc_pc(*x,y,z*), sc_pcs_in_ph(*x,y*), scs_pcs()
*caec or casc.set_eq_xx Where **xx** can take:	P(*i*),T(*i*),H(*i*),G(*i*),S(*i*), Cp(*i*),V(*i*),VT(*i*)
*caec or casc.get_eq_xx Where **xx** can take:	T(),P(),V(),VT(),VM(), S(),A(),H(),G(),X_,IA_ ,A_,AC_,CP_,CPM_, G_,GM_ ,H_ ,HM_,MU_,S_, SM_,V_,VM_