DOCUMENTATION: INTERFACES

Note: The asterisk indicates that the argument is optional.

- **Transient 1D reactive transport class**:

Subroutine	Input Arguments			Description
	Name	Kind	Description	
`read_time_discretisation`	`unit`	Integer	Unit of the file to be read	This subroutine reads the file ending in "_WMA_discr.dat" that contains the time discretization for the WMA: number of time steps, time step(s) and method integration of chemical reactions.
	`root`	String	Path of the file to be read	
`write_RT_1D`	`unit`	Integer	Unit of the file where the results are to be written	This subroutine writes the data and results in the final time of all variables to the file ending in ".out".
	`root`	String	Path of the file where the results are to be written	
	`path_py*`	String	Path of the directory where the results needed to use them in Python are to be written	
`set_transport`	`transport_obj`	Transient 1D transport class	Transient 1D transport class object.	Assigns the `transport` attribute of this class.
`set_chemistry`	`chemistry_obj`	Chemistry class	Chemistry class object	Assigns the `chemistry` attribute of this class.

- **Transient 1D transport class**:

`read_transport_data_WMA`	`unit`	Integer	The unit of the file to be read	This subroutine reads the file ending in "_WMA_lambdas.dat" that contains the porosity, mixing ratios, and water indices corresponding to the mixing ratios.
	`root`	String	Root of the file to be read	

- **Chemistry Class**:

`set_option`	`option`	Integer	Chemical Information Reading Option	This subroutine assigns the value of the `option` attribute belonging to the chemical class. This attribute dictates the format that is to be used to read the chemical information. `option=1`: based on CHEPROO (only operating format right now) `option=2`: PHREEQC. `option=3`: PFLOTRAN.
`set_Jac_flag`	`Jac_flag`	Integer	Jacobian computation model	This subroutine assigns the value of the `Jac_flag` attribute belonging to the chemistry class. This attribute dictates the model to be used to calculate all Jacobians. `Jac_flag=0`: Incremental coefficients `Jac_flag=1`: Analytically
`read_chemistry`	`root`	String	contains the root of the data entry files of the problem to be solved	This subroutine reads all the chemical information. It first calls the chemical system's reading subroutine via its attribute of the chemical system class, then reads the types of initial waters, solids, and gases, and finally reads the target waters association. In addition, it assigns objects in the *Reactive Zone* class belonging to the *Chemistry* class.
	`path_DB`	String	contains the directory of chemical databases	
	`unit_chem_syst_file`	Integer	The unit of the file containing the chemical system (ending in "_sist_quim.dat")	
	`unit_loc_chem_file`	Integer	Unit of the file containing the initial types of waters, solids, and gases (ending in "_quim_loc.dat")	
	`unit_target_waters_file`	Integer	Unit of the file containing the association between Target Waters and initial water, solid, and	

			gas types (ending in "_tar_wat.dat")	
	unit_output_file	Integer	Unit of the file containing the options for writing results (ending in "_out_options.dat")	
solve_reactive_mixing	mixing_ratios	Real Array Class	Mixing Ratios	This is the main reactive mix solver. Performs the reactive mixing calculations with the WMA for a given time interval. At the local level, the calculations are made by the *Aqueous Chemistry* class.
	mixing_waters_indices	Integer Array Class	Mixing water indices corresponding to mixing ratios	
	F_mat	Real Vector	Storage matrix	
	time_discr	Time discretization class	Time discretization used to solve transport	
	int_method_chem_reacts	Integer	Temporal integration method for chemical reactions. 1: Euler Explicit 2: Euler Fully Implicit	

Sequence in the main file

1) Define path names: problem to be solved, databases and Python (optionally)
2) Define file units
3) Choose problem to solve. If it is not among the options, it must be added.
4) Initialize transport:
    a. Transient 1D Transport Class Object `my_tpt_trans` calls read_transport_data_WMA
    b. Transient 1D Reactive Transport Class Object `my_RT_trans` calls `read_time_discretisation`
    c. `my_RT_trans` calls for `set_transport` using `my_tpt_trans`
5) Initialize chemistry:
    a. Set chemical reading option
    b. Chemical class object `my_chem` calls `set_option`
    c. Define Jacobian computation model
    d. `my_chem` calls `set_Jac_flag`
    e. `my_chem` calls `read_chemistry`
6) Solve reactive mixing and assign chemistry class object:
    a. `my_chem` calls `solve_reactive_mixing`
    b. `my_RT_trans` calls set_chemistry using my_chem
7) Write data and results:
    a. `my_RT_trans` calls write_RT_1D