```
!!
!!     ____            _           _                    ___ _
!!    |  _ \ _ __ ___ (_) ___  ___| |_   ___ ___  _ __ / __(_) __ _
!!    | |_) | '__/ _ \| |/ _ \/ __| __| / __/ _ \| '_ \| |_| |/ _` |
!!    |  __/| | | (_) | |  __/ (__| |_ | (_| (_) | | | |  _| | (_| |
!!    |_|   |_|  \___// |\___|\___|\__(_)___\___/|_| |_|_| |_|\__, |
!!                  |__/                                       |___/
!!
!! PyRED - http://q4md-forcefieldtools.org
!! Compatible with Python versions 2.5 up to 2.7.18
!!
!!           The Project.config file
!! http://q4md-forcefieldtools.org/REDServer-Development/Documentation/Project.config
!!   Description of the keywords used by PyRED
!!         Documentation of February 2015
!!       Last update of this documentation:
!!              March 25th, 2025
!!
!! Always reload this file, when reading this file with a web browser to be sure to
!! access to the latest version!
!!
!! The Project.config file is optional.
!! It is provided as an input within the archive file to overwrite default tasks.
!! See http://q4md-forcefieldtools.org/Tutorial/Mini-HowTo-InputFiles.pdf
!!     http://q4md-forcefieldtools.org/Tutorial/Tutorial-4.php
!!
!! Keywords provided in the 'Project.config' file are related to the input molecules
!! provided by the user in the PyRED job.
!!
!! If an error is made in a keyword, the PyRED job is executed with the default value.
!!
!! The Project.config file only contains plain text (no rich text file format).
!! It has to be prepared or modified using a text editor such as the vi, gedit, nedit
!! geany, or the notepad.
!!
!!    FF   = empirical force field
!!    lib. = library
!!    QM   = quantum mechanics
!!    MEP  = molecular electrostatic potential
!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!                                                                                  !!
!!   A KEYWORD IN THE PROJECT.CONFIG FILE MUST BE ACTIVATED TO BE USED BY PyRED:    !!
!! A KEYWORD IS ACTIVATED BY DELETING THE '#' CHARACTER AT THE BEGINNING OF A LINE  !!
!! ACTIVATE A KEYWORD ONLY WHEN NECESSARY AND AFTER FULLY UNDERSTANDING ITS MEANING !!
!!                                                                                  !!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!
!! In a PyRED job, which involves multiple orientations multiple conformations
!!   and/or multiple molecules:
!!   - 'n' is the molecule index involving multiple molecules
!!   - each molecule can contain 'c' conformation(s)
!!   - each conformation can be involved in a 'o' reorientation procedure
!!  'n', 'c' and 'o' are integers, which are strickly greater than '0'.
!!     i. e. n > 0;  c > 0;  o > 0...
!!
!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!        Molecule keywords, which depend on the 'n' molecule index        !!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!
!! The title of molecule 'n':
```

```
!! The MOLECULE'n'-TOTCHARGE keyword allows defining an informative title for
!!   a molecule. If this keyword is not provided the MOLECULE-'n' title is
!!   assigned by default.
!! This tile is used to identify the molecule(s) involved in FF generation in
!!   the output files generated by PyRED
!! Do not use any blank character in the title, and replace the blank character
!!   by the underscore '_' character
!! The title of the Mol_red1.pdb input molecule is g+/g+_conformation_dimethylphosphate:
# MOLECULE1-TITLE         = g+/g+_conformation_dimethylphosphate
!! The title of the Mol_red2.pdb input molecule is Methanol:
# MOLECULE2-TITLE         = Methanol
# MOLECULE3-TITLE         = ...
!!
!! The total charge of molecule 'n':
!! The MOLECULE'n'-TOTCHARGE keyword is required only if the total charge
!!   value of the molecule does not equal '0'. If this keyword is not provided,
!!   0 is the total charge value assigned by default.
!! The molecule total charge can also be determined from the PDB input file, see:
!! http://q4md-forcefieldtools.org/REDServer-Development/Documentation/readme.txt
!! The total charge of the Mol_red1.pdb input molecule is -1; this is an anion:
# MOLECULE1-TOTCHARGE    = -1
!! The total charge of the Mol_red2.pdb input molecule is 1; this is a cation:
# MOLECULE2-TOTCHARGE    = 1
# MOLECULE3-TOTCHARGE    = ...
!!
!! The spin multiplicity of molecule 'n':
!! The MOLECULE'n'-SPINMULT keyword is required only if the spin multiplicity
!!   value of the molecule does not equal '1'. If this keyword is not provided,
!!   1 is the spin multiplicity value assigned by default.
!! See the examples below to determine the spin multiplicity of a molecule
!! Spin multiplicity = 0 is not possible!
!! Set spin multiplicity = 1 for a molecule without single electron:
!!   (most of the cases for bioorganic molecules)
!! No single electron (unpaired) in the Mol_red1.pdb input file:
# MOLECULE1-SPINMULT     = 1
!! One single electron in the Mol_red2.pdb input file:
!! Example Cu(II): nine electrons in the five d orbitals
!!              a single electron in the five d orbitals:
# MOLECULE2-SPINMULT     = 2
!! Two single electrons in the Mol_red3.pdb input file:
!! Example: https://en.wikipedia.org/wiki/Triplet_oxygen .O-O.
# MOLECULE3-SPINMULT     = 3
!! Five single electrons in the Mol_red3.pdb input file:
!! Example Fe(III): five single electrons in the five d orbitals:
!! See http://q4md-forcefieldtools.org/Tutorial/Tutorial-4-demo21.pdf
# MOLECULE3-SPINMULT     = 6
!! No single electron in the Mol_red3.pdb input file:
!! Example Zn(II): ten paired electrons in the five d orbitals:
# MOLECULE3-SPINMULT     = 1
!!
!! The atoms of the 'n' input molecule(s) are automatically reordered so that
!!   the hydrogen atoms are located after the heavy atom they are bound to
!! This option is usefull to make atom indexes more 'human understandable',
!!   when keywords related to the input atom order have to be defined by the user
!!   (geometrical and/or charge constraints, atom typing, etc...):
!!   In that case, one has to proceed in two steps:
!! - a first job is executed, where the PDB input files are simply reordered
!!   (atom indexes can be graphically determined in the dedicated JSmol applet
!!   generated in each R.E.D.Server Development job)
!! - a second productive job is then run, where the reordered PDB input files
!!   (i. e. the optimized PDB outputs from the previous job) associated with
!!   keywords with more intuitive (and less error prone) atom indexes are
!!   provided by the user
!! If not provided MOLECULE'n'-ATMREORDR = ON is the default.
!! If MOLECULE'n'-ATMREORDR = OFF the atom order provided in the PDB input
!!   file is conserved.
!! Reorder the atoms of the Mol_red1.pdb input molecule according to the algorithm
!!   developed in PyRED:
# MOLECULE1-ATMREORDR    = ON
!! Do not reorder the atoms of the Mol_red2.pdb input molecule:
# MOLECULE2-ATMREORDR    = OFF
# MOLECULE3-ATMREORDR    = ...
!!
!! Automatically correct the atom names of molecule 'n' so that two atoms in
!!   a residue cannot share the same name: indeed to be differentiated, two
!!   atoms belonging to a residue cannot have the same name in a FF lib (.mol2
!!   file)
!! See http://q4md-forcefieldtools.org/Tutorial/Tutorial-4.php#Bpol2
!! If not provided MOLECULE'n'-ATMCOR = ON1 is the default.
!! If MOLECULE'n'-ATMCOR = ON1 only redundant atom names are renamed.
!! If MOLECULE'n'-ATMCOR = ON2 all the atoms are renamed so that an atom name
!!   is composed by its element and an integer, that is incremented.
!! If MOLECULE'n'-ATMCOR = OFF atom names are not corrected, even if
!!   duplicates are found (not recommended).
# MOLECULE1-ATMCOR       = ON1
# MOLECULE2-ATMCOR       = ON2
# MOLECULE3-ATMCOR       = ...
!!
!! Atom connectivities allow defining the molecular topology i. e. the single
!!   bonds in the FF lib. (mol2 file), and are used in charge equivalencing and
!!   charge derivation
!! Atom connectivities are automatically determined by PyRED from each optimized
!!   geometry: atom connectivities provided in the PDB input file through the
!!   PDB CONECT keyword are not read by default
!! If not provided MOLECULE'n'-CALCONECT = ON is the default.
!! Setting MOLECULE'n'-CALCONECT = OFF, and providing the atom connectivities
!!   in the PDB input file with the CONECT keyword prevent automatic atom
!!   connectivity determination. This approach is advantageous for bioinorganic
!!   complexes (with a metal center), where long single bonds have to be defined
!!   See: http://q4md-forcefieldtools.org/Tutorial/Tutorial-4.php#II4
!! Using MOLECULE'n'-CALCONECT = OFF is not recommended for organic molecules,
!!   and is even error-prone: to be used with care
# MOLECULE1-CALCONECT    = ON
# MOLECULE2-CALCONECT    = OFF
# MOLECULE3-CALCONECT    = ...
!!
!! Chemical equivalencing used in the charge fitting step is automatically
!!   determined for molecule 'n':
!! If not provided MOLECULE'n'-CALCHEMEQ = ON is the default.
!! Set MOLECULE1-CALCHEMEQ = OFF to cancel chemical equivalencing (not
!!   recommended).
# MOLECULE1-CALCHEMEQ    = ON
# MOLECULE2-CALCHEMEQ    = ON
# MOLECULE3-CALCHEMEQ    = ...
!!
!! Automatically correct the residue names of molecule 'n' in PyRED outputs:
!! Atoms are commonly grouped within a residue, and a residue name is generally
!!    composed by three (more rarely four) characters.
!! If not provided MOLECULE'n'-RESMOD = ON is the default.
!! If MOLECULE'n'-RESMOD = ON, a single residue name by molecule is generated,
!!   and atom names are corrected accordingly. The use of this keyword is
!!   strongly advised: fundamentally, generating a FF lib. composed of different
!!   residues is correct, but the LEaP program from the Amber tools is only able
!!   to load FF lib. composed of a single residue, when one wants to get matches
!!   with a file obtained from the protein data bank:
!!   See http://q4md-forcefieldtools.org/Tutorial/Tutorial-4.php#Bpol2
!! If MOLECULE'n'-RESMOD = OFF and if they are judged correct residue names
!!   of molecule 'n' are conserved.
# MOLECULE1-RESMOD       = ON
# MOLECULE2-RESMOD       = OFF
# MOLECULE3-RESMOD       = ...
!!
!! Definition of the atom names of molecule 'n':
!! If this keyword is not provided the atom names are automatically determined
!!   from the PDB input file.
!! The atom name order in that keyword matters; it is that in the PDB input files:
!! The atom name order is that in the Mol_red1.pdb input file:
# MOLECULE1-ATMNAME      = C1 H11 H12 H13 O3' P O1P O2P O5' C2 H21 H22 H23
!!              (atom names for dimethylphosphate are taken as an example here)
!! The atom name order is that in the Mol_red2.pdb input file:
# MOLECULE2-ATMNAME      = C1 H11 H12 H13 OH HO
!!              (atom names for methanol are taken as an example here)
!!
!! Definition of the chemical elements of the atoms of molecule 'n'
!! If this keyword is not provided the elements are automatically determined
!!   from the PDB input file.
!! The element order in that keyword matters; it is that in the PDB input files:
!! The element order is that in the Mol_red1.pdb input file:
# MOLECULE1-ATMELME      = C H H H O P O O O C H H H
!!       (chemical elements of dimethylphosphate are taken as an example here)
!! The element order is that in the Mol_red2.pdb input file:
# MOLECULE2-ATMELME      = C H H H O H
!!       (chemical elements of methanol are taken as an example here)
!!
!! Definition of the atom types of molecule 'n':
!! Useful to modify the atom types automatically determined by PyRED, and/or to
!!   define new atom types: the atom types in a FF lib. (mol2 file) have to be
!!   graphically checked in the dedicated JSmol applet generated in each R.E.D.
!!   Server Development job, and can be modified using this keyword in a new
!!   PyRED job
!! When new atom types are defined, a 'frcmod.user' file can also be provided as
!!   input to PyRED to provide the FF parameters related to these new atom types
!! If this keyword is not provided the atom types are automatically determined
!!   based on the dictionary of atom types developed in PyRED
!! The atom type order in that keyword matters; it is that in the PDB input files:
!! The atom type order is that in the Mol_red1.pdb input file:
# MOLECULE1-ATMTYPE      = CT H1 H1 H1 OS P  O2 O2 OS CT H1 H1 H1
!! The atom type order is that in the Mol_red2.pdb input file:
# MOLECULE2-ATMTYPE      = CT H1 H1 H1 OH HO
!! GAFF atom types can be used in association with the 'gaff.dat' file provided
!!   as a 'frcmod.user' input file:
# MOLECULE1-ATMTYPE      = c3 h1 h1 h1 os p5 o  o  os c3 h1 h1 h1
!!                (atom types of dimethylphosphate are taken as an example here)
# MOLECULE2-ATMTYPE      = c3 h1 h1 h1 oh ho
!!                (atom types of methanol are taken as an example here)
!! Atom types of other force fields can also be provided here:
!! GAFF, GAFF2, Amber14FFSB, Amber19FFSB, GLYCAM 2006, OPLS...
!! See http://q4md-forcefieldtools.org/Tutorial/Tutorial-4-demo5.pdf
!!     http://q4md-forcefieldtools.org/Tutorial/Tutorial-4-demo6.pdf
!! Reminder:
!! An atom type generally consists of one or two alphanumeric characters.
!! Special characters, such as the asterisk (*), may also be used.
!! PyRED also handles atom types with three alphanumeric characters. In that case,
!! check whether the MD software used supports three-character atom types.
!!
!! Forcing charge/chemical equivalencing of molecule 'n' for the charge fitting
!!   step carried out without intra-molecular charge constraint(s):
!! This is achived by providing atom names based on two empirical/basic rules:
!!   (i) atoms, which are chemically equivalent bear the same element and the
!!   same integer,
!!   (ii) if one wants to recompute the charge values of atoms or groups of
!!   toms in the second resp stage (usually the charges of the methylene and
!!   methyl groups) the heavy atoms of these groups have to be underlined using
!!   the '@' character.
!! Chemical/charge equivalencing (and charges values) in a FF lib. (mol2 file)
!!   has to be graphically checked in the dedicated JSmol applet generated in
!!   each R.E.D. Server Development job, and charge equivalencing can be
!!   modified using this keyword in a new PyRED job
!! If not provided chemical equivalencing is automatically determined.
!! The atom order in that keyword matters; it is that in the PDB input files:
!! The atom order is that in the Mol_red1.pdb input file:
# MOLECULE1-CHEMEQSM     = C@1 H1 H1 H1 O1 P1 O2 O2 O1 C@1 H1 H1 H1
!!           (empirical rules for dimethylphosphate are taken as an example here)
!!           (dimethylphosphate: the two methyl groups are made equivalent)
!! The atom order is that in the Mol_red2.pdb input file:
# MOLECULE2-CHEMEQSM     = C@1 H1 H1 H1 O2 H2
!!           (empirical rules for methanol are taken as an example here)
!!
!! Forcing charge/chemical equivalencing of molecule 'n' for the charge fitting
!!   step carried out with intra-molecular charge constraint(s):
!! This is achived by providing specific atom names; see empirical rules above
!! If not provided chemical equivalencing is automatically determined.
!! This keyword is often required when using intra-molecular charge constraint
!!   and can be adapted from the MOLECULE'n'-CHEMEQSM keyword printed in the
!!   'Project-out.config' file obtained from a previous PyRED job (see the
!!   'Data-Default-Proj' directory).
!! The atom order in that keyword matters; it is that in the PDB input files:
!! The atom order is that in the Mol_red1.pdb input file:
# MOLECULE1-CHEMEQIA     = C1 H2 H3 H4 O1 P1 O5 O5 O6 C@7 H7 H7 H7
!!             (dimethylphosphate: first methyl group involved in an INTRA-MCC1)
!! The atom order is that in the Mol_red2.pdb input file:
# MOLECULE2-CHEMEQIA     = C@1 H1 H1 H1 O2 H2
!!             (methanol: OH and HO atoms are involved in an INTRA-MCC1)
!!
!! Forcing charge/chemical equivalencing of molecule 'n' for the charge fitting
!!   step involving multiple molecules:
!! This is achived by providing specific atom names; see empirical rules above
!! If not provided chemical equivalencing is automatically determined.
!! This keyword is often required for the charge fitting step, which involves
!!   multiple molecules, and can be adapted from the MOLECULE'n'-CHEMEQSM keyword
!!   printed in the 'Project-out.config' file obtained from a previous PyRED job
!!   (see the 'Data-Default-Proj' directory).
!! The atom order in that keyword matters; it is that in the PDB input files:
!! The atom order is that in the Mol_red1.pdb input file:
# MOLECULE1-CHEMEQMM     = C1 H2 H3 H4 O5 P6 O7 O7 O8 C9 H10 H11 H12
!!             (dimethylphosphate: two methyl groups involved in two INTER-MCC1)
!! The atom order is that in the Mol_red2.pdb input file:
# MOLECULE2-CHEMEQMM     = C@1 H1 H1 H1 O2 H2
!!             (empirical rules for methanol are taken as an example here)
!!
!! The rigid body re-orientation algorithm (RBRA) is applied to each optimized
!!   geometry/conformation of molecule 'n' before MEP computation:
!! This allows deriving highly reproducible MEP-based charge values.
!!   see http://www.ncbi.nlm.nih.gov/pmc/articles/PMC2918240/
!! Two sets of indexes for 3 non-linear atoms are automatically selected from
!!   the center of mass of the molecule, and involved in the RBRA procedure.
!!   A 'o' = 2 molecular reorientation MEP-based computation is consequently
!!   carried out (recommended). In that case the two set of indexes are chosen
!!   so that the impact of the first reorientation on the MEP is cancelled out
!!   by the second one.
!! If not provided MOLECULE'n'-RBRA = ON is the default.
!! If MOLECULE'n'-RBRA = OFF the molecular orientation generated after
!!   geometry optimization and defined by the QM program is directly involved
!!   in MEP  computation. A 'o' = 1 MEP-based computation is carried out, and
!!   leads to non-reproducible charge values (not recommended).
!!   see http://www.ncbi.nlm.nih.gov/pmc/articles/PMC2918240/
# MOLECULE1-RBRA         = ON
# MOLECULE2-RBRA         = OFF
# MOLECULE3-RBRA         = ...
!!
```

```
!! If MOLECULE'n'-RBRA = ON the procedure controlled using the MOLECULE'n'-
!!   REORIENT keyword allows managing reorientations (i.e. a translation
!!   and a series of rigid body rotations) for each optimized geometry/
!!   conformation of molecule 'n'.
!!   see http://www.ncbi.nlm.nih.gov/pmc/articles/PMC2918240/
!! Set(s) of indexes for 3 non-linear atoms are defined in MOLECULE'n'-REORIENT
!!   and involved in the procedure.
!! If MOLECULE'n'-RBRA = ON and none of the three MOLECULE'n'-REORIENT,
!!   MOLECULE'n'-ROTATE and MOLECULE'n'-TRANSLATE keywords is provided two
!!   sets of 3 non-linear atoms are automatically selected by PyRED (see
!!   above; recommended).
!! Atom indexes are these in the Mol_red1.pdb input file:
!! The optimized geometry, which is reoriented is Mol_m1/Mol_m1-c1-qmra.pdb
!! The optimized geometry of the Mol_red1.pdb input file is reoriented four times
!!   and four MEP are generated:
# MOLECULE1-REORIENT     = 1 2 3 | 3 2 1 | 1 6 10 | 10 6 1
!! Atom indexes are these in the Mol_red2.pdb input file:
!! The optimized geometry, which is reoriented is Mol_m2/Mol_m2-c1-qmra.pdb
!! The optimized geometry of the Mol_red2.pdb input file is reoriented twice
!!   and two MEP are generated:
# MOLECULE2-REORIENT     = 1 2 3 | 3 2 1
!! Atom indexes are these in the Mol_red3.pdb input file:
# MOLECULE3-REORIENT     = ...
!!
!! If MOLECULE'n'-RBRA = ON the procedure controlled using the MOLECULE'n'-
!!   ROTATE keyword allows managing rotations (i.e. a series of rigid body
!!   rotations) for each optimized geometry/conformation of molecule 'n'.
!! Set(s) of indexes for 3 non-linear atoms are defined in MOLECULE'n'-ROTATE
!!   and involved in the procedure.
!! Using the same indexes in MOLECULE'n'-ROTATE or in MOLECULE'n'-REORIENT
!!   leads to identical charge values.
!! If MOLECULE'n'-RBRA = ON and none of the three MOLECULE'n'-REORIENT,
!!   MOLECULE'n'-ROTATE and MOLECULE'n'-TRANSLATE keywords is provided two
!!   sets of indexes for 3 non-linear atoms are automatically selected.
!! Atom indexes are these in the Mol_red1.pdb input file:
!! The optimized geometry, which is rotated is Mol_m1/Mol_m1-c1-qmra.pdb
!! The optimized geometry of the Mol_red1.pdb input file is rotated four times
!!   and four MEP are generated:
# MOLECULE1-ROTATE       = 1 2 3 | 3 2 1 | 1 6 10 | 10 6 1
!! Atom indexes are these in the Mol_red2.pdb input file:
!! The optimized geometry, which is rotated is Mol_m2/Mol_m2-c1-qmra.pdb
!! The optimized geometry of the Mol_red2.pdb input file is reoriented twice
!!   and two MEP are generated:
# MOLECULE2-ROTATE       = 1 2 3 | 3 2 1
!! Atom indexes are these in the Mol_red3.pdb input file:
# MOLECULE3-ROTATE       = ...
!!
!! If MOLECULE'n'-RBRA = ON the procedure controlled using the MOLECULE'n'-
!!   TRANSLATE keyword allows managing translations carried out on the X, Y
!!   and Z axes for each optimized geometry/conformation of molecule 'n'.
!! A translation results in different set of Cartesian coordinates used in
!!   MEP computation, but does not affect charge values.
!! Set(s) of 3 distances are defined in the MOLECULE'n'-TRANSLATE keyword,
!!   and applied on the X, Y and Z axes.
!! If MOLECULE'n'-RBRA = ON and none of the three MOLECULE'n'-REORIENT,
!!   MOLECULE'n'-ROTATE and MOLECULE'n'-TRANSLATE keywords is provided two
!!   sets of 3 non-linear atoms are automatically selected by PyRED
!! The optimized geometry, which is translated is Mol_m1/Mol_m1-c1-qmra.pdb:
!! The optimized geometry of the the Mol_red1.pdb input file is translated
!!   four times and four MEP are generated:
!!   - Add +1 to the X axis for the first translation
!!   - Add +2 to the Y axis for second translation
!!   - Add -3 to the Z axis for the third translation
!!   - Add 1, 2 and -3 to the X, Y and Z axes for the fourth translation,
!!     respectively
```

```
# MOLECULE1-TRANSLATE    = 1  0  0 | 0  2  0 | 0  0 -3 | 1  2 -3
!! The optimized geometry, which is translated is Mol_m2/Mol_m2-c1-qmra.pdb:
!! The optimized geometry of the the Mol_red2.pdb input file is translated
!!   twice and two MEP are generated:
# MOLECULE2-TRANSLATE    = 1  0  0 | 0  2  0
!! Atom indexes are these in the Mol_red3.pdb input file:
# MOLECULE3-TRANSLATE    = ...
!!
!! Request the use of an intra-molecular charge constraint for molecule 'n'
!!   during the charge fitting step:
!! This allows constraining the charge value of an atom or a group of atoms
!!   as well as generating molecular fragment(s) from molecule 'n'.
!!   see https://upjv.q4md-forcefieldtools.org/Tutorial/Tutorial-4.php#IV1
!! The format to be used is the following:
!!   Value of the constraint | atom indexes belonging to the constraint |
!!   R flag ('Remove' the atoms involved in the constraint in the FF lib.)
!!   or K flag ('Keep' the atoms involved in the constraint in the FF lib.)
!! INTRA-MCC1: the intra-molecular charge constraint is not repeated in the
!!   second fitting stage (if any), and/or the charge values of chemically
!!   equivalent atoms, that belong to the list of atoms involved in the
!!   constraint, are not equivalenced. This leads to the use of a minimum
!!   number of constraints during the charge fitting step: this is the
!!   approach that was originally developed, and that one generally needs:
!!   indeed, atoms involved in intra-molecular charge constraint are generally
!!   removed from the FF lib.: thus, they do not need to be equivalenced
!!   (because not involved in molecular dynamics simulations).
!! INTRA-MCC2: the intra-molecular charge constraint is repeated in the second
!!   fitting stage (if any), and/or the charge values of chemically equivalent
!!   atoms, that belong to the list of atoms involved in the constraint, are
!!   equivalenced. This approach uses additional constraints for the charge fit
!!   and should be avoided (except in some rare cases if one masters charge fit).
!!   Indeed, increasing the number of constraints leads to an increase of the
!!   RRMS value, which reflects a more important error generated for the
!!   charge fit.
!! More generally, if you do not understand the differences between the INTRA-MCC1
!!   and INTRA-MCC2 keywords, just use INTRA-MCC1.
!! Atom indexes are these in the Mol_red1.pdb input file:
# MOLECULE1-INTRA-MCC1   =  0.0 | 1 2 3 4 5 6 | Remove
# MOLECULE1-INTRA-MCC1   =  0.15 | 1 2 3 4 | Remove
# MOLECULE1-INTRA-MCC1   = -0.15 | 1 2 3 4 5 6 7 8 | Remove
!! Atom indexes are these in the Mol_red2.pdb input file:
# MOLECULE2-INTRA-MCC1   = -0.5064 | 8 | Keep
# MOLECULE2-INTRA-MCC1   =  1.2098 | 9 | Keep
!!  or
!! Atom indexes are these in the Mol_red1.pdb input file:
# MOLECULE1-INTRA-MCC2   = 0.0 | 1 2 3 4 5 6 7 | R
!!                             (to be avoided or used with care!)
# MOLECULE1-INTRA-MCC2   = 0.0 | 1 2 3 4 5 6 7 | K
!!
!! Request the definition of a head for the first open valency created for
!!   a molecular fragment(s) built from molecule 'n':
!! If not provided a head is the default for the first open valency of a
!!   fragment
!! If MOLECULE'n'-MOL3HEAD = OFF a tail is defined for the first open
!!   valency of a fragment.
!! The definition of a 'head' or a 'tail' for a molecular fragment only
!!   matters when the mol3 lib. file format is selected.
!!   see http://q4md-forcefieldtools.org/Tutorial/leap-mol3.php
!! See the definition of the 'head' and the 'tail' of a molecular fragment
!!   from the documentation of the LEaP program. See:
!! http://q4md-forcefieldtools.org/REDServer-Development/Documentation/AmberTools12.pdf#Objects
!! http://q4md-forcefieldtools.org/REDServer-Development/Documentation/AmberTools12.pdf#desc
# MOLECULE1-MOL3HEAD     = ON
# MOLECULE2-MOL3HEAD     = OFF
# MOLECULE3-MOL3HEAD     = ...
```

```
!!
!! Request the generation of lone-pair(s) (LP) on an heteroatom (HA) such
!!   as an oxygen, nitrogen or sulfur atom based on the work of
!!   Dixon & Kollman J. Comput. Chem. 1997, 18, 1632-1646.
!! Default algorithm:
!! MOLECULE1-LONEPAIR = HA_index
!! Atom indexes are these in the Mol_red1.pdb input file:
# MOLECULE1-LONEPAIR = 5
# MOLECULE1-LONEPAIR = 9
!! Atom indexes are these in the Mol_red2.pdb input file:
# MOLECULE2-LONEPAIR = 5
# MOLECULE2-LONEPAIR = 9
!! Modification of the default algorithm i.e. modification of the distance
!!   and angle values (in angstroms and degrees) as well as the number of lone
!!   pair(s), respectively; to be provided on a single line:
!! MOLECULE1-LONEPAIR = HA_index  Distance_HA-LP  Atom1_index
!!    Angle_value1_Atom1-HA-LP  Atom2_index  Angle_value2_Atom2-HA-LP
!!    | Number_of_LP
!!
!! Automatic generation of the amino acid fragments (N-terminal, C-terminal
!!   and central fragments) from a dipeptide molecule:
!! Simply provide the atom indexes of the two capping groups (generally the
!!   CH3CO and NHCH3 groups) of the dipeptide molecule.
!! This keyword can only be used for the following type of dipeptide analogs:
!!   CH3CO-NHCH(side-chain)CO-NHCH3
!! Atom indexes are these in the Mol_red1.pdb input file:
# MOLECULE1-FRGAA    = 1 2 3 4 5 6 | 20 21 22 23 24 25
!! Atom indexes are these in the Mol_red2.pdb input file:
# MOLECULE2-FRGAA    = 1 2 3 4 5 6 | 20 21 22 23 24 25
!!
!! Automatic generation of the nucleotide fragments (5'-terminal, 3'-terminal
!!   and central fragments) from a nucleoside molecule:
!! Simply provide the atom indexes of the two groups of atoms (generally the
!!   HO3' and HO5' hydroxyl groups) involved in the phosphodiester backbone.
!! This keyword can only be used for nucleoside analogs, which contains two
!!   terminal hydroxyl groups
!! Atom indexes are these in the Mol_red1.pdb input file:
# MOLECULE1-FRGNT    = 1 2 | 3 4
!! Atom indexes are these in the Mol_red2.pdb input file:
# MOLECULE2-FRGNT    = 1 2 | 3 4
!!
!! Definition of geometrical constraints during geometry optimization:
!! Positional constraint: provide the atom index to be constrained (fixed).
!!   To be repeated for each atom to be constrained:
!! Atom indexes are these in the Mol_red1.pdb input file:
# MOLECULE1-GEOMOPTCONST = 1
# MOLECULE1-GEOMOPTCONST = 10
!! Bond constraint: provide the two atom indexes involved in the
!!   constrained bond | bond value (in angstroms - optional)
!!   To be repeated for each bond to be constrained:
!! Atom indexes are these in the Mol_red1.pdb input file:
# MOLECULE1-GEOMOPTCONST = 1 2 | 1.5
!! Angle constraint: provide the three atom indexes involved in the
!!   constrained angle | angle value (in degrees - optional)
!!   To be repeated for each angle to be constrained:
!! Atom indexes are these in the Mol_red1.pdb input file:
# MOLECULE1-GEOMOPTCONST = 1 2 3 | 109.5
!! Dihedral constraint: provide the four atom indexes involved in the
!!   constrained dihedral | dihedral value (in degrees - optional)
!!   To be repeated for each dihedral to be constrained:
!! Atom indexes are these in the Mol_red1.pdb input file:
# MOLECULE1-GEOMOPTCONST = 1 2 3 4 | -60.0
!! Important remarks:
!! Using a positional constraint for an atom is not possible in GAMESS or
!!   Firefly.
!! When using Gaussian 03, Gaussian 09 A.02, GAMESS or Firefly bond, angle
!!   and dihedral values are read. However, it is advised to provide
!!   Cartesian coordinates close to the constrained value(s). This allows
!!   faster convergence.
!! When using Gaussian 09 (ver C.01, D.01 or E.01) and Gaussian 16 (B.01,
!!   C.01) bond, angle and dihedral values are not read. Thus, the user must
!!   provide Cartesian coordinates in agreement with the constrained value(s).
!!   Using a PDB input file with four digits after the decimal point increases
!!   the accuracy.
!! QM geometry optimization is performed without any geometrical constraint
!!   by default.
!! Geometrical constraint(s) may be used during QM geometry optimization to
!!   prevent the formation of unwanted non-bonding interactions: indeed,
!!   geometry optimization carried out in gas phase tends to overestimate
!!   non-bonding interactions such as hydrogen-bonds or salt-bridges compared
!!   to a computation performed in condensed phases.
!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!     Molecule keywords, which does not depend on the molecule index     !!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!
!! Request the definition of a single residue in each molecular fragment
!!   generated. Atoms are automatically renamed if duplicates are found.
!! If not provided FRAGMENT-RESMOD = ON is the default.
!! If FRAGMENT-RESMOD = OFF the residues in the built fragments are based on
!!   the residues defined in the input molecule(s).
# FRAGMENT-RESMOD        = ON
!!
!! Request the use of an inter-molecular charge constraint between two
!!   different molecules involved in a multiple molecules job:
!! This allows generating molecular fragment(s) from two molecules.
!! See https://upjv.q4md-forcefieldtools.org/Tutorial/Tutorial-4.php#IV2
!!     https://upjv.q4md-forcefieldtools.org/Tutorial/Tutorial-4.php#IV3
!!     https://upjv.q4md-forcefieldtools.org/Tutorial/Tutorial-4.php#IV4
!! The format is the following:
!! Value of the constraint | two molecule indexes involved in this constraint
!! | atom indexes belonging to the first molecule involved in the constraint
!! | atom indexes belonging to the second molecule involved in the constraint
!! INTER-MCC1: the constraint is not repeated in the second fitting stage,
!!   and/or the charge values of chemically equivalent atoms, that belong to
!!   the list of atoms involved in the constraint, are not equivalenced.
!!   This leads to the use of a minimum number of constraints during the
!!   charge fitting step, and is the approach that was originally developed.
!! INTER-MCC2: the constraint is repeated in the second fitting stage and/or
!!   the charge values of chemically equivalent atoms, that belong to the
!!   list of atoms involved in the constraint, are equivalenced.
!! INTER-MCC1 vs INTER-MCC2: see the documentation above about INTRA-MCC1.
!! In case of doubt, just use INTER-MCC1.
# MOLECULE-INTER-MCC1    = 0.0 | 1 2 | 10 11 12 13 | 1 2
# MOLECULE-INTER-MCC1    = 0.0 | 1 2 |  1  2  3  4 | 3 4
!! No MOLECULE index in the 'MOLECULE-INTER-MCC1' keyword;
!! Molecule indexes are provided in the second field of the keyword between
!! two pipe characters ('1 2' in the example above)
!! Above the sum of charge values of atom indexes '10 11 12 13' of molecule '1'
!! plus the sum of charge values of atom indexes '1 2' of molecule '2' are forced
!! to equal 0.0.
!! Also the sum of charge values of atom indexes '1 2 3 4' of molecule '1'
!! plus the sum of charge values of atom indexes '3 4' of molecule '2' are forced
!! to equal 0.0.
!!  or
# MOLECULE-INTER-MCC2    = 0.0 | 1 2 | 10 11 12 13 | 1 2
# MOLECULE-INTER-MCC2    = 0.0 | 1 2 |  1  2  3  4 | 3 4
!!                             (to be avoided or used with care!)
!!
!! Request the use of inter-molecular charge equivalencing between atoms
!!   belonging to different molecules involved in a multiple molecules job:
!!   i.e. charge values of atoms belonging to identical groups of atoms in
!!   different molecules are forced to be equivalent.
!! Two formats are implemented: MOLECULE-INTER-MEQA and MOLECULE-INTER-MEQB:
!! INTER-MEQA: Molecule indexes involved in the constraints | identical atom
!! indexes belonging to the molecules involved in the equivalencing
# MOLECULE-INTER-MEQA    = 2 3 4 5 | 1 2 3 4
!! No MOLECULE index in the 'MOLECULE-INTER-MEQA' keyword;
!! Molecule indexes are provided in the first field of the keyword between
!! the equal and pipe characters ('2 3 4 5' in the example above)
!! Charge values of atom indexes '1' of molecules '2 3 4 5' are equivalenced,
!! charge values of atom indexes '2' of molecules '2 3 4 5' are equivalenced,
!! charge values of atom indexes '3' of molecules '2 3 4 5' are equivalenced, and
!! charge values of atom indexes '4' of molecules '2 3 4 5' are equivalenced.
!! INTER-MEQB: Molecule indexes involved in the constraints | list(s) of atom
!! indexes involved in the constraints separated by the '-' character
!! (different atom indexes may be used)
# MOLECULE-INTER-MEQB    = 2 3 4 | 10 10 10 - 11 11 11 - 22 23 24 - 33 34 35
!! No MOLECULE index in the 'MOLECULE-INTER-MEQB' keyword
!! Molecule indexes are provided in the first field of the keyword between
!! the equal and pipe characters ('2 3 4' in the example above)
!! Charge values of atom indexes '10 10 10' of molecules '2 3 4' are equivalenced,
!! charge values of atom indexes '11 11 11' of molecules '2 3 4' are equivalenced,
!! charge values of atom indexes '22 23 24' of molecules '2 3 4' are equivalenced, and
!! charge values of atom indexes '33 34 35' of molecules '2 3 4' are equivalenced.
!!
!! Control of the radius of the [K - LR] elements used in MEP computation
!! This option is limited to the execution of the Gaussian program.
!! - If defined in Bondi J. Phys. Chem. 1964, 68, 441 the Bondi radii are used.
!! - If not defined by A. Bondi and not provided by using the keyword below 1.8
!!   is the value used by default.
!! Providing ELEMENT-RAD4MEP keyword(s) allows overwriting default values used
!!   in MEP computation
# FE-RAD4MEP = 1.7
!!
!! Control of the radii of the [H - LR] elements to define the atom
!!   connectivities, i. e. the topology of the molecule(s) in the FF libraries
!! Providing ELEMENT-RAD4TOP keyword(s) allows overwriting default values used
!!   by PyRED, and modifying the molecular topology(ies) involved in atom typing
!!   and FF parameter generation.
!! This type of keywords can be particularly useful for metal atoms, i. e. if
!!   the default values used by PyRED does not lead to the wanted topology for a
!!   molecule.
!! See http://q4md-forcefieldtools.org/Tutorial/Tutorial-4.php#II4
# LI-RAD4TOP = 1.5 or 1.0 (vs the 1.28 default value)
# NA-RAD4TOP = 2.0 or 1.0 (vs 1.66)
# K-RAD4TOP  = 2.5 or 1.5 (vs 2.03)
# FE-RAD4TOP = 2.0, 1.5 or 1.0 (vs 1.32)
!!
# MOLECULE-NA_ALLDATA   = OFF/ON
!! For nucleosides strictly follow the Amber atom naming convention in the PDB
!!  input files, and set MOLECULE-NA_ALLDATA = ON in the Project.config file:
!! Default distances, angles, dihedral angles, hydrogen-bonds and five-membered
!!  ring pucker parameters (amplitude of pucker & phase angle of pseudorotation)
!!  are automatically calculated from QM optimized geometries using predefined
!!  atom names (no need to use the MOLECULE-NA_DISTANCE, MOLECULE-NA_ANGLE,
!!  MOLECULE-NA_DIHEDRAL, MOLECULE-NA_HBOND and MOLECULE-NA_PUCKER keywords)
# MOLECULE-NA_DISTANCE  = OFF/ON
!! List-Distances.inf is the file, where the distances/bonds to be computed are
!!  listed: provide the two atom names (separated by a space character) involved
!!  in the distance to be calculated on the same line
# MOLECULE-NA_ANGLE     = OFF/ON
!! List-Angles.inf is the file, where the angles to be computed are listed:
!!  provide the three atom names (separated by a space character) involved in
!!  the angle to be calculated on the same line
```

```
# MOLECULE-NA_DIHEDRAL  = OFF/ON
!! List-Dihedrals.inf is the file, where the dihedral angles to be computed are
!!  listed: provide the four atom names (separated by a space character) involved
!!  in the dihedral angle to be calculated on the same line
# MOLECULE-NA_HBOND     = OFF/ON
!! List-Hbonds.inf is the file, where the intra-molecular hydrogen bonds to be
!!  evaluated are listed: provide the three atom names (separated by a space
!!  character) involved in the hydrogen bond to be evaluated on the same line
!!  as it follows: Donor_atom-name Hydrogen_atom-name Acceptor_atom-name
# MOLECULE-NA_PUCKER    = OFF/ON
!! List-Dihedrals4pucker.inf is the file, where the dihedral angles used to
!!  calculate the five-membered ring pucker parameters are listed: provide the
!!  four atom names (separated by a space character) of the five dihedral angles
!!  in five different lines (order: lines 1 to 5: nu0 to nu4) to compute the
!!  amplitude of pucker Tm, and the phase angle of pseudorotation P
!! Proline pyrrolidine ring is also handled (use the N CA CB CG CD atom names),
!!  and provide the 5 chi1-chi5 dihedral angles in the List-Dihedrals4pucker.inf
!!  file
!! OFF is the default for the MOLECULE-NA_ALLDATA, MOLECULE-NA_DISTANCE,
!!     MOLECULE-NA_ANGLE, MOLECULE-NA_DIHEDRAL, MOLECULE-NA_HBOND and
!!     MOLECULE-NA_PUCKER keywords
!! The List-Distances.inf, List-Angles.inf, List-Dihedrals.inf, List-Hbonds.inf,
!!  and List-Dihedrals4pucker.inf files have to be included in the archive file
!!  to be uploaded, if needed (comments may be added in these txt files, but each
!!  space character has to be replaced by the underscore one)
!! The bond, angle, dihedral angle, hydrogen bond and pucker geometrical features
!!  are printed in the Measure-Geometry.log file available after clicking on the
!!  'Geometrical features' link available in the project graphical interface.
!!
```