

OpenEye
SCIENTIFIC

OEFF TK – C#

Release 2026.1.0

Cadence Design Systems, Inc.

August 06, 2026

CONTENTS

1 Theory **3**
- 1.1 Components . . . 3
  - 1.1.1 Objective Functions . . . 3
  - 1.1.2 Molecule Objective Function . . . 3
  - 1.1.3 Adaptors . . . 3
  - 1.1.4 Optimizers . . . 3
- 1.2 Force Field . . . 3
  - 1.2.1 MMFF . . . 3
  - 1.2.2 FF14SB . . . 5
  - 1.2.3 SMIRNOFF . . . 5
  - 1.2.4 AMBER force field terms . . . 5
  - 1.2.5 MMFFAmber . . . 7
  - 1.2.6 MMFFIEFF . . . 7
- 1.3 Sheffield Model . . . 7
  - 1.3.1 Small molecules . . . 7
  - 1.3.2 Proteins . . . 8

2 OEFF Examples **9**
- 2.1 Simple Functions and Optimization . . . 9
  - 2.1.1 Solving a simple equation . . . 9
  - 2.1.2 Using checkpoints for optimization monitoring . . . 9
- 2.2 Molecule Functions . . . 9
  - 2.2.1 User defined molecule function . . . 9
- 2.3 Small Molecule Force fields . . . 9
  - 2.3.1 Single ligand in vacuum . . . 9
  - 2.3.2 Energy components of a single ligand in vacuum . . . 9
  - 2.3.3 MMFF interactions of a single ligand . . . 10
- 2.4 Protein-Ligand Complexes . . . 10
  - 2.4.1 Protein-ligand optimization . . . 10
- 2.5 Using adaptors . . . 10
  - 2.5.1 Optimizing a single ligand with fixed atoms . . . 10
  - 2.5.2 Optimizing a single ligand torsions . . . 10
  - 2.5.3 Optimizing rigid ligand in protein . . . 10
- 2.6 Custom Force Fields . . . 10
  - 2.6.1 Custom Smirnoff . . . 10

3 OEFF API **11**
- 3.1 OEOpt Classes . . . 11
  - 3.1.1 OEAdaptor . . . 11
  - 3.1.2 OEBFGSOpt . . . 12

3.1.3 OECGLineMinimize . . . 13
3.1.4 OECheckpoint0 . . . 14
3.1.5 OECheckpoint1 . . . 15
3.1.6 OEConjGradOpt . . . 15
3.1.7 OEDFPOpt . . . 16
3.1.8 OEFComponent . . . 17
3.1.9 OEFletcherReevesOpt . . . 17
3.1.10 OEFunc0 . . . 17
3.1.11 OEFunc1 . . . 20
3.1.12 OEFunc2 . . . 23
3.1.13 OEHestenesStiefelOpt . . . 25
3.1.14 OELineMinimize . . . 26
3.1.15 OEMaxStepLineMinimize . . . 27
3.1.16 OENRLineSearch . . . 27
3.1.17 OEOptimizer0 . . . 28
3.1.18 OEOptimizer1 . . . 28
3.1.19 OEOptimizer2 . . . 30
3.1.20 OEQNLineMinimize . . . 31
3.1.21 OERibierePolakOpt . . . 31
3.1.22 OESteepestDescentOpt . . . 32
3.1.23 OENewtonOpt . . . 33
3.2 OEMolPotential Classes . . . 33
3.2.1 OEBendParams . . . 33
3.2.2 OEBendPotential . . . 35
3.2.3 OECoulombPotential . . . 36
3.2.4 OEFFParams . . . 37
3.2.5 OEFFPotential . . . 43
3.2.6 OEForceField . . . 44
3.2.7 OEGenericFF . . . 47
3.2.8 OEGenericFF2 . . . 49
3.2.9 OEInteraction . . . 51
3.2.10 OEInteractParams . . . 52
3.2.11 OEInterAdaptor . . . 53
3.2.12 OEInterCoulombPotential . . . 55
3.2.13 OEInterMolFunc1 . . . 56
3.2.14 OEInterMolFunc2 . . . 58
3.2.15 OEInterNonBondPotBase . . . 59
3.2.16 OEInterNonBondPotential . . . 60
3.2.17 OEInterVdwPotential . . . 61
3.2.18 OEMolAdaptor . . . 62
3.2.19 OEMolAdaptor2 . . . 64
3.2.20 OEMolFunc . . . 65
3.2.21 OEMolFunc0 . . . 69
3.2.22 OEMolFunc1 . . . 72
3.2.23 OEMolFunc2 . . . 75
3.2.24 OENonBondPotBase . . . 76
3.2.25 OENonBondPotential . . . 78
3.2.26 OENonBondIntcsOptions . . . 79
3.2.27 OENumericMolFunc2 . . . 80
3.2.28 OEOutOfPlaneParams . . . 82
3.2.29 OEOutOfPlanePotential . . . 83
3.2.30 OEQuatAdaptor . . . 84
3.2.31 OEScaledMolFunc . . . 86
3.2.32 OEStretchParams . . . 88

3.2.33 OEStretchPotential . . . 89
3.2.34 OESubsetAdaptor . . . 90
3.2.35 OETorAdaptor . . . 92
3.2.36 OETorQuatAdaptor . . . 94
3.2.37 OETorsionParams . . . 96
3.2.38 OETorsionPotential . . . 98
3.2.39 OEVdwParams . . . 99
3.2.40 OEVdwPotential . . . 100
3.3 OEMolPotential Constants . . . 101
3.3.1 OEFFParam . . . 101
3.3.2 OEVarType . . . 105
3.3.3 OEFuncType . . . 106
3.4 OEConstr Classes . . . 108
3.4.1 OEDihedralQuadPenalty . . . 108
3.4.2 OEHarmonicPotential . . . 110
3.5 OEMolMMFF Classes . . . 111
3.5.1 OEMMFF . . . 111
3.5.2 OEMMFF94sParams . . . 113
3.5.3 OEMMFFOutOfPlane . . . 114
3.5.4 OEMMFFParams . . . 116
3.5.5 OEMMFFStretchBend . . . 117
3.6 OEMolMMFF Functions . . . 118
3.6.1 OEIsValidMMFFMolecule . . . 118
3.6.2 OEMMFFCalcVdw . . . 119
3.7 OEMolSmirnoff Classes . . . 119
3.7.1 OEParsley . . . 119
3.7.2 OEParsleyParams . . . 120
3.7.3 OESage . . . 121
3.7.4 OESageParams . . . 122
3.7.5 OESmirnoff . . . 123
3.7.6 OESmirnoffParams . . . 124
3.7.7 AddBendData . . . 126
3.7.8 AddImproperData . . . 126
3.7.9 AddStretchData . . . 126
3.7.10 AddTorsionData . . . 126
3.7.11 AddVdWData . . . 127
3.7.12 GetBendData . . . 127
3.7.13 GetImproperData . . . 127
3.7.14 GetStretchData . . . 127
3.7.15 GetTorsionData . . . 127
3.7.16 GetVdWData . . . 128
3.7.17 Load . . . 128
3.7.18 SetBendData . . . 128
3.7.19 SetImproperData . . . 128
3.7.20 SetStretchData . . . 128
3.7.21 SetTorsionData . . . 128
3.7.22 SetVdWData . . . 128
3.7.23 OESmirnoffData . . . 129
3.7.24 OESmirnoffStretchData . . . 129
3.7.25 OESmirnoffBendData . . . 130
3.7.26 OESmirnoffTorsionData . . . 131
3.7.27 OESmirnoffImproperData . . . 133
3.7.28 OESmirnoffVdWData . . . 134
3.8 OEMolSmirnoff Constants . . . 135

3.8.1 OESmirnoffType . . . 135
3.9 OESheff Classes . . . 135
3.9.1 OESheffield . . . 135
3.9.2 OESheffieldOptions . . . 137
3.10 OESheff Constants . . . 139
3.10.1 OESheffieldParamType . . . 139
3.11 OEFF Classes . . . 139
3.11.1 OEFF14SBParsley . . . 139
3.11.2 OEFF14SBParsleyComplex . . . 141
3.11.3 OEFF14SBParsleyParams . . . 141
3.11.4 OEFF14SBSage . . . 143
3.11.5 OEFF14SBSageComplex . . . 144
3.11.6 OEFF14SBSageParams . . . 145
3.11.7 OEFF14SBSmirnoff . . . 146
3.11.8 OEFF14SBSmirnoffComplex . . . 147
3.11.9 OEFF14SBSmirnoffParams . . . 148
3.11.10 OEComplexEnergies . . . 149
3.11.11 OEComplexFF . . . 151
3.11.12 GetLigandFF . . . 153
3.11.13 GetVar . . . 153
3.11.14 Set . . . 153
3.11.15 SetHostFlex . . . 153
3.11.16 SetNonBondCutoff . . . 153
3.11.17 SetupHost . . . 154
3.11.18 OEComplexFFParameter . . . 154
3.11.19 OEForceFieldParameter . . . 155
3.11.20 OEMMFFIEFF . . . 156
3.11.21 OEMMFFAmber . . . 158
3.11.22 OEMMFFAmberComplex . . . 159
3.11.23 OEMMFFAmberOptions . . . 160
3.11.24 OEMMFFComplex . . . 163
3.11.25 OEMMFFIEFFOptions . . . 164
3.11.26 OEMMFFSheffield . . . 165
3.11.27 OEMMFFSheffieldOptions . . . 166
3.11.28 OESmirnoffComplex . . . 168
3.12 OEFF Functions . . . 169
3.12.1 OEClearForceFieldDummyAtom . . . 169
3.12.2 OEClearForceFieldDummyAtoms . . . 169
3.12.3 OEGetMMFFSheffieldFFName . . . 169
3.12.4 OEGetMMFFSheffieldFFType . . . 170
3.12.5 OEIsForceFieldDummyAtom . . . 170
3.12.6 OEIsValidMMFFSheffieldFF . . . 170
3.12.7 OESetForceFieldDummyAtom . . . 171
3.12.8 OEFFGetArch . . . 171
3.12.9 OEFFGetLicensee . . . 171
3.12.10 OEFFGetPlatform . . . 171
3.12.11 OEFFGetRelease . . . 172
3.12.12 OEFFGetSite . . . 172
3.12.13 OEFFGetVersion . . . 172
3.12.14 OEFFIsLicensed . . . 172
3.13 OEFF Constants . . . 172
3.13.1 OELigandFFType . . . 172
3.13.2 OEMMFFSheffieldFFType . . . 173

4 Release History **175**
4.1 OEFF TK 2.9.2 . . . 175
4.1.1 Minor Bug Fixes . . . 175
4.2 OEFF TK 2.9.1 . . . 175
4.3 OEFF TK 2.9.0 . . . 175
4.3.1 New Features . . . 175
4.3.2 Minor Bug Fixes . . . 176
4.4 OEFF TK 2.8.0 . . . 176
4.4.1 New features . . . 176
4.4.2 Minor Bug Fixes . . . 176
4.5 OEFF TK 2.7.1.3 . . . 176
4.6 OEFF TK 2.7.1 . . . 176
4.6.1 New features . . . 176
4.6.2 Minor bug fixes . . . 177
4.7 OEFF TK 2.7.0 . . . 177
4.7.1 New features . . . 177
4.7.2 Major bug fixes . . . 177
4.7.3 Minor bug fixes . . . 177
4.8 OEFF TK 2.5.2.1 . . . 177
4.9 OEFF TK 2.5.2.0 . . . 178
4.9.1 Major bug fixes . . . 178
4.10 OEFF TK 2.5.1 . . . 178
4.10.1 New features . . . 178
4.10.2 Major bug fixes . . . 178
4.10.3 Minor bug fixes . . . 178
4.11 OEFF TK 2.5.0 . . . 178
4.11.1 New features . . . 178
4.11.2 Major bug fixes . . . 179
4.12 OEFF TK 2.4.0 . . . 179
4.12.1 New features . . . 179
4.12.2 Minor bug fixes . . . 179
4.13 OEFF TK 2.3.1 . . . 179
4.14 OEFF TK 2.3.0 . . . 179
4.14.1 New features . . . 180
4.14.2 Documentation changes . . . 181
4.15 OEFF TK 2.2.0 . . . 181
4.15.1 New features . . . 181
4.15.2 Major bug fixes . . . 182
4.15.3 Minor bug fixes . . . 182
4.16 OEFF TK 2.1.0 . . . 183
4.16.1 New features . . . 183
4.16.2 Major bug fixes . . . 183
4.16.3 Documentation changes . . . 183
4.17 OEFF TK 2.0.4 . . . 184
4.18 OEFF TK 2.0.3 . . . 184
4.18.1 New features . . . 184
4.18.2 Minor bug fixes . . . 184
4.18.3 Documentation changes . . . 184
4.19 OEFF TK 2.0.2 . . . 184
4.20 OEFF TK 2.0.1 . . . 185
4.20.1 New features . . . 185
4.20.2 Minor Bug Fixes . . . 185
4.21 OEFF TK 2.0.0 . . . 186
4.21.1 Licensing . . . 186

4.21.2 OEFF functionality . . . 186
4.21.3 Objective functions and molecular interactions . . . 186
4.21.4 Force fields . . . 186
4.21.5 Optimizers . . . 187
4.21.6 Adaptors . . . 187
4.21.7 New features . . . 187

5 Copyright and Trademarks **189**

6 Sample Code **191**

7 Citation **193**
7.1 Orion® . . . 193
7.2 Orion Floes . . . 193
7.3 Toolkits and Applications . . . 193
7.4 Publications for Bibliographies . . . 195
7.4.1 Orion . . . 195
7.4.2 AFITT and FLYNN Applications . . . 195
7.4.3 OEDocking Application and Toolkit . . . 195
7.4.4 OMEGA Application and Toolkit . . . 195
7.4.5 ROCS Application . . . 196
7.5 OpenEye MMDS Web Service . . . 196

8 Technology Licensing **197**
8.1 GCC . . . 269
8.1.1 GCC RUNTIME LIBRARY EXCEPTION . . . 270
8.1.2 GNU GENERAL PUBLIC LICENSE . . . 271

Index **285**

Attention

OEFF TK is currently only available in C++ and Python.

CHAPTER

ONE

THEORY

1.1 Components

1.1.1 Objective Functions

At the core of the objective functions are pure virtual interfaces *OEFunc0*, *OEFunc1* and *OEFunc2*. A new objective function can be defined by providing appropriate expressions for these interface methods. The objective functions can be used along with the optimizers to find the minima of any function.

1.1.2 Molecule Objective Function

The molecule objective functions extend the objective function interfaces to define molecular interactions through *OEMolFunc*. Similar to the objective functions, a new molecule objective function can also be defined by providing the appropriate expressions for the interface methods. All of the force field components are defined as molecule objective functions to allow easy minimization of molecule systems.

1.1.3 Adaptors

Adaptors wrap molecule objective functions to allow working in different coordinate systems, instead of the default all atom Cartesian coordinates. Adaptors can be powerful tool for selective energy evaluation and minimization of specific subdomains. Methods to convert between the adaptor coordinates and the Cartesian coordinates are also provided through these interfaces.

1.1.4 Optimizers

A number of general purpose optimizers are provided that can be used to minimize any basic or extended objective functions. Optimizer interfaces are defined through *OEOptimizer1* and *OEOptimizer2*. Pros and cons of various optimizers are established in the literature, and one needs to use their understanding of the problem in choosing the appropriate optimizer. Optimizers that can solve objective functions with known gradients but no Hessian (*OEOptimizer1*) requires an additional line minimizer (*OELineMinimize*).

1.2 Force Field

1.2.1 MMFF

The MMFF potential expression is:

$$V_{MMFF} = \sum_b V_b + \sum_a V_a + \sum_s V_s + \sum_o V_o + \sum_t V_t + \sum_v V_v + \sum_c V_c$$

where the seven terms respectively describe bond stretching (b), angle bending (a), stretch-bending (s), out-of-plane bending (o), torsion (t), Van der Waals (v) and electrostatic (c) interactions. Their functional forms are given below.

Bond stretching

For a bond *b* between atoms *i* and *j* the stretching potential is:

$$V_b = 143.9325\frac{k_{ij}}{2}\Delta r_{ij}^2(1 + c_s\Delta r_{ij} + 7/12c_s^2\Delta r_{ij}^2)$$

where k_{ij} is the force constant, Δr_{ij} is the difference between actual and reference bond lengths, and c_s is so called "cubic-stretch" constant for which the value is -2 $Å^{-1}$.

Angle bending

The bending potential of a bond angle *a* made by the bonds between atoms *i*, *j* and atoms *j*, *k* is given by:

$$V_a = 0.043844\frac{k_{ijk}}{2}\Delta\vartheta_{ijk}^2(1 + c_b\Delta\vartheta_{ijk})$$

where k_{ijk} is the force constant, $\Delta\vartheta_{ijk}$ is the difference between actual and reference bond angles, and c_b is the so called "cubic-bend" constant for which the value is $-0.4rad^{-1}$.

Stretch-bend interaction

The coupling between the stretching potential of two bonds forming a bond angle and bending that angle is described by:

$$V_s = 2.5121(k_{ijk}\Delta r_{ij} + k_{kji}\Delta r_{kj})\Delta\vartheta_{ijk}$$

where k_{ijk} and k_{kji} are force constants which couple stretches of *i-j* and *k-j* to the *i-j-k* bend respectively. Δr and $\Delta\vartheta$ are defined above.

Out-of-plane bending

For a trigonal center *j*, the potential of displacement for an atom *l* bonded to atom *j* out of plane *i-j-k* is:

$$V_o = 0.043844\frac{k_{ijkl}}{2}\chi_{ijkl}^2$$

where k_{ijkl} is the force constant and χ_{ijkl} is an angle formed by the bond *j-l* and the plane *i-j-k*.

Torsion interactions

For every four bonded atoms *i-j-k-l* the torsion interaction is described by the term:

$$V_t = 0.5(V_1(1 + \cos\Phi) + V_2(1 - \cos 2\Phi) + V_3(1 + \cos 3\Phi)$$

where V_1, V_2 and V_3 are constants depending on atoms *i*, *j*, *k*, *l*, and Φ is the dihedral angle formed by bonds *i-j* and *k-l*.

Van der Waals interactions

For a pair of atoms *i* and *j* separated by three or more bonds, where the distance between them is r_{ij}, MMFF adopts the following Van der Waals potential:

$$V_v = \epsilon_{ij}\left(\frac{1.07R_{ij}}{r_{ij} + 0.07R_{ij}}\right)^7\left(\frac{1.12R_{ij}^7}{r_{ij}^7 + 0.12R_{ij}} - 2\right)$$

where R_{ij} and ϵ_{ij} are defined as follows:

$$R_i = A_i\alpha_i^{0.25}$$

$$R_{ij} = 0.5(R_i + R_j)(1 + B(1 - \exp(-12\gamma_{ij}^2)))$$

$$\gamma_{ij} = (R_i - R_j)/(R_i + R_j)$$

$$\epsilon_{ij} = \frac{181.16 G_i G_j \alpha_i \alpha_j}{(\alpha_i/N_i)^{0.5} + (\alpha_j/N_j)^{0.5}} \frac{1}{R_{ij}^6}$$

where α_i is atomic polarizability of atom *i*, *B* is 0.2 or 0.0 if one of the atoms is polar hydrogen, N_i and N_j are the Slater-Kirkwood effective numbers of valence electrons, G_i, G_j and A_i are scale factors.

Electrostatic interactions

The electrostatic interaction between two charged atoms *i* and *j* separated by at least three bonds is calculated from the standard Coulombic expression:

$$V_c = f \frac{332.0716 q_i q_j}{D(r_{ij} + \delta)}$$

where *D* is the dielectric constant for which the default value is 1, q_i and q_j are the MMFF partial charges on atoms *i* and *j*, r_{ij} is the interatomic distance, δ is the "electrostatic buffering" constant of 0.05 Å. Scaling factor f is 0.75 for 1,4 interactions, and 1.0 otherwise.

1.2.2 FF14SB

FF14SB is Amber primary force field for protein modeling. Its details are described in the paper of Maier et al. ([Maier-2015]) and in http://ambermd.org/AmberModels.php. The functional form of FF14SB is described under the chapter "AMBER force field terms".

1.2.3 SMIRNOFF

SMIRNOFF (SMIRKS Natural Open Force Field) is a kind of small-molecule force field springing from the Open Force Field Initiative (OFF Initiative or OFFI) (https://openforcefield.org/). The key feature of this force field is its chemical representation of parameters by using extended SMARTS patterns. "Extended" here means that the SMARTS patterns also use atom assignment syntax from SMIRKS. For example, `[*:2]~[*:1]~[#1]` specifies three atoms, where the first is assigned to be atom 2, the second atom is assigned to be atom 1, and the third atom is not assigned. This format frees the force field from using atom types, instead it uses the direct chemical perception of the extended SMARTS patterns ([Mobley-2018]). Compared to traditional atom typing approaches, this representation greatly reduces the number of parameters needed to specify a general small-molecule force field and makes each parameter more independent.

While it currently has a traditional "Amber" type functional form as specified below, future versions expect to modify the functional form. The OFF Initiative will be releasing periodic revisions of the parameters which will include changes to the chemical specifications (i.e. the extended SMARTS patterns) as well as the numerical parameters. The revision used by the API points will be given with the results.

Currently SMIRNOFF force field has 6 similar to those in AMBER force field and are given below.

1.2.4 AMBER force field terms

Amber force field functional form has 6 terms:

$$V_{AMBER} = \sum_b V_b + \sum_a V_a + \sum_o V_o + \sum_t V_t + \sum_v V_v + \sum_c V_c$$

describing respectively bond stretching (*b*), angle bending (*a*), out-of-plane displacement (*o*), torsion (*t*), Van der Waals (*v*) and electrostatic (*c*) interactions.

Bond stretching

For *a* bond *b* the stretching potential has a simple harmonic form:

$$V_b = k_b(r - r_0)^2$$

where k_b is the force constant, r, and r_0 are actual and reference bond length, respectively.

Angle bending

For bond angle *a*, the angle bending potential form is:

$$V_a = k_a(\alpha - \alpha_0)^2$$

where k_a is the force constant, α, and α_0 are actual and reference bond angle values, respectively.

Torsion interactions

Every 4 bonded atoms defined a torsion whose energy is represented with 1 to 6 terms of a Fourier series:

$$V_t = \sum_n V_n/d_n(1 + \cos(p_n\theta - \gamma_n))$$

where V_n, d_n, p_n and γ_n are barrier height, degeneracy, periodicity and phase for the *n* component of a torsion whose dihedral angle is θ

Out-of-plane bending

It is defined as potential of a displacement of the trigonal center atom *a* bonded to 3 other out-of-plane atoms, *b*, *c* and *d*. Its functional form is similar to a regular torsion with a single term. Sometimes it is also called an *improper torsion*.

$$V_o = V(1 + \cos(p\theta - \gamma))$$

While in FF14SB this term is used to preserved planarity of the sp2 configuration, any improper torsion moving a central trigonal atom out of plane can be used, in SMIRNOFF this term has been redefined to account for the fact that one can define 3 improper torsions of moving a trigonal atom *a* out of plane defined by atoms *b*, *c* and *d*. The energy of such a displacement in SMIRNOFF is calculated as 1/3 of the sum of those torsions.

Van der Waals interactions

For a pair of nonbonded atoms *i* and *j*, SMIRNOFF force field adopts Lennard-Jones potential:

$$V_v = \epsilon_{ij}\left[\left(\frac{R_{ij}}{r_{ij}}\right)^{12} - 2\left(\frac{R_{ij}}{r_{ij}}\right)^{6}\right]$$

where ϵ_{ij} is well depth, R_{ij} and r_{ij} are the equilibrium and actual distance between atoms *i* and *j*, respectively.

Electrostatic interactions

A simple Coulomb term:

$$V_c = \frac{q_i q_j}{4\pi\epsilon_0 r_{ij}}$$

represents Coulomb interactions, where q_i and q_j are partial atomic charges, ϵ_0 is the vacuum permittivity, and r_{ij} is the distance between atoms *i* and *j*.

1.2.5 MMFFAmber

Intermolecular Amber potential

Intermolecular interaction can be described by the Amber force field instead of MMFF potential:

$$V_{Amber} = \sum_{i,j} \left\{ \epsilon_{ij} \left[\left(\frac{R_{ij}}{r_{ij}} \right)^{12} - 2 \left(\frac{R_{ij}}{r_{ij}} \right)^{6} \right] + \frac{q_i q_j}{4\pi\epsilon_0 r_{ij}} \right\}$$

where the summation is over protein-ligand atom pairs. r_{ij} is the interatomic distance, R_{ij} is the VdW distance for a pair of atoms, q_i and q_j are the Amber partial charges on atoms *i* and *j*, and ϵ_0 is the vacuum permittivity. VdW parameters R_{ij} and ϵ_{ij} and partial charges are taken from work by Wang and coworkers ([Wang-J-2000]).

1.2.6 MMFFIEFF

Intermolecular IEFF potential

Intermolecular interaction can be described by the IEFF (Intermolecular Force Field) instead of MMFF potential.

The IEFF is an ongoing development effort internally at **OpenEye** to obtain high accuracy Intermolecular Force Field. The initial development of IEFF was done during 2010. The goal of the initial development was to prove internally at OpenEye as well as to external collaborators that point charge monopole approximation in the Coulomb terms are inadequate to capture intermolecular Coulomb interactions within chemical accuracy (within about 1 kcal/mol) regardless of the charge models used. The higher accuracy in the Coulomb terms is especially important and vital in optimizing and ranking organic crystal structures. The initial IEFF development has had some successful crystal predictions. It has also clearly shown, for the formamide dimer case, that there is a 1D energy curve along a given torsion angle where all tested point charge models fails even qualitatively showing repulsive interactions while accurate ab-initio calculations and IEFF with multipole based Coulomb interactions show a minimum along the torsion with great agreement to each other.

The IEFF has not been improved significantly since those initial academic studies and cannot be considered as a general purpose intermolecular force field due to lack of sufficient coverage. However, major development effort is underway to make IEFF an industry quality general intermolecular force field.

The currently available IEFF supports only a few selected elements that were the most important ones for organic crystal predictions, including H, C, N, O, S, F, and Cl. The atomic multipoles are obtained by an internal analysis program that obtains the SCF converged density matrices from ab-initio (PSI4) calculations. The current IEFF does not use any atom types, all VdW parameters are associated to each element types. This obviously introduces a very significant weakness especially for hydrogens where both polar and apolar hydrogens share the same FF parameters but it is also important deficiency for other elements as well. The new version of the IEFF that is under development will try to address all of these deficiencies.

1.3 Sheffield Model

1.3.1 Small molecules

The "Sheffield Solvation Model" [Grant-2007] is a simple two parameter model for calculating the electrostatic component of molecular solvation energy. The Sheffield model is robust and works well for calculating solvation energies of small drug-like molecules with good accuracy, when compared against the more accurate Poisson–Boltzmann calculations, and requires significantly low computational resource. The model is represented by the following simple expression:

$$E_{Sheff} = -\frac{f_\epsilon}{4\pi\epsilon_0} \frac{1}{2} \sum_{i,j} \frac{Q_i Q_j}{\sqrt{A\sigma_i\sigma_j + BR_{ij}^2}}$$

where f_ϵ is

$$f_\epsilon = \frac{1}{\epsilon_{in}} - \frac{1}{\epsilon_{out}}$$

and ϵ_{in}, ϵ_{out} are he relative permittivities (dielectric constants) in solute and solvent, respectively. Q_i and Q_j are charges on atoms i and j, σ_i and σ_j are corresponding atomic radii, and R_{ij} is an interatomic distance between atoms i and j. A and B are parameters to the model.

This model works well in the situation where atoms in molecules are well exposed to the solvent. In macromolecules like proteins, this condition is not met.

1.3.2 Proteins

In order to apply Sheffield model for proteins and complexes a modification is required to correct the atomic radii according to their exposure to solvent. The correction to atomic radii is applied by introduction of a third parameter to the original two parameter model.

$$\sigma_i = \sigma_{0,i} + C\left(1 - \sqrt{\frac{S_i}{S_{0,i}}}\right)$$

where S_i and $S_{0,i}$ are atomic contributions to the solvent accessible surface of atom i in the protein and in a residue to which it belongs which was isolated from the rest of the protein. Initial atomic radii $\sigma_{0,i}$ was assumed to be of ZAP9 [Nicholls-2010] type.

CHAPTER

TWO

OEFF EXAMPLES

2.1 Simple Functions and Optimization

2.1.1 Solving a simple equation

The following example illustrates how to define a simple objective function. By deriving the objective function from *OEFunc2*, we can find the roots of the simple quadratic equation using *OENewtonOpt* optimizer. A class derived from *OEFunc2* must contain analytical gradients and Hessians. This example expects a number as input for the initial guess to solve the simple function.

2.1.2 Using checkpoints for optimization monitoring

The following example illustrates how to define checkpoints and use then along with optimizers to monitor progress during optimization. For simplicity, a simple quadratic equation is defined as objective function and derived from *OEFunc1*. The quadratic equation is solved using the *OEBFGSOpt* optimizer. A class derived from *OEFunc1* must contain analytical gradients. This examples expects a number as input for the initial guess to solve the simple function.

2.2 Molecule Functions

2.2.1 User defined molecule function

The following example illustrates how to define an objective function within the context of a molecule. Generally speaking, a molecule function (*OEMolFunc*) defines some sort of interaction involving a part or all of a molecule. For simplicity, a simple energy function is defined that consists sum of square of distance of all the atoms in the molecule. The molecule function is optimized using the *OEBFGSOpt* optimizer. A class derived from *OEMolFunc1* must contain analytical gradients.

2.3 Small Molecule Force fields

2.3.1 Single ligand in vacuum

The following example illustrates how to calculate energy and optimize a single ligand in vacuum. The molecule needs to be prepared (`OEForceField.PrepMol`), followed by a call to `OEMolFunc.Setup` to create the interactions. Optimization is carried out using the *OEBFGSOpt* optimizer.

2.3.2 Energy components of a single ligand in vacuum

The following example illustrates how to obtain various intermolecular energy components of a single ligand in vacuum.

2.3.3 MMFF interactions of a single ligand

The following example illustrates how to obtain various interactions of a single ligand in the context of a force field.

2.4 Protein-Ligand Complexes

2.4.1 Protein-ligand optimization

The following example illustrates how to set up a ligand optimization within the context of a protein contained in a design unit.

2.5 Using adaptors

2.5.1 Optimizing a single ligand with fixed atoms

The following example illustrates how to fix a subset of atoms using the *OESubsetAdaptor* during a single ligand optimization in vacuum. The adaptor is initialized with the force field interactions of the ligand, and passed as the objective function to be optimized. Methods of the adaptor are used to convert between the Cartesian coordinates of the ligand and the adaptor variables.

2.5.2 Optimizing a single ligand torsions

The following example illustrates how to optimize a set of torsion angles using the *OETorAdaptor* for a single ligand in vacuum. The adaptor is initialized with the force field interactions of the ligand, and passed as the objective function to be optimized. Methods of the adaptor are used to convert between the Cartesian coordinates of the ligand and the adaptor variables.

2.5.3 Optimizing rigid ligand in protein

The following example illustrates how to perform rigid optimization of a ligand in the context of a protein using the *OEQuatAdaptor*. The adaptor is initialized with the protein-ligand interactions, and passed as the objective function to be optimized. Methods of the adaptor are used to convert between the Cartesian coordinates of the ligand and the adaptor variables.

2.6 Custom Force Fields

2.6.1 Custom Smirnoff

The following example illustrates how to load custom SMIRNOFF force field parameters from an OFFXML file and use that to calculate energy and optimize a single ligand in vacuum.

CHAPTER

THREE

OEFF API

3.1 OEOpt Classes

3.1.1 OEAdaptor

> **Attention**
>
> This API is currently available in C++ and Python.

```
class OEAdaptor
```

The *OEAdaptor* class is a pure virtual base class which defines the interface for classes which adapt or transform variables and gradients during an optimization. Adaptors are useful in cases where function evaluations are performed in a different coordinate system than the variables being optimized (for example polar coordinates are optimized but the function is evaluated in Cartesian coordinates).

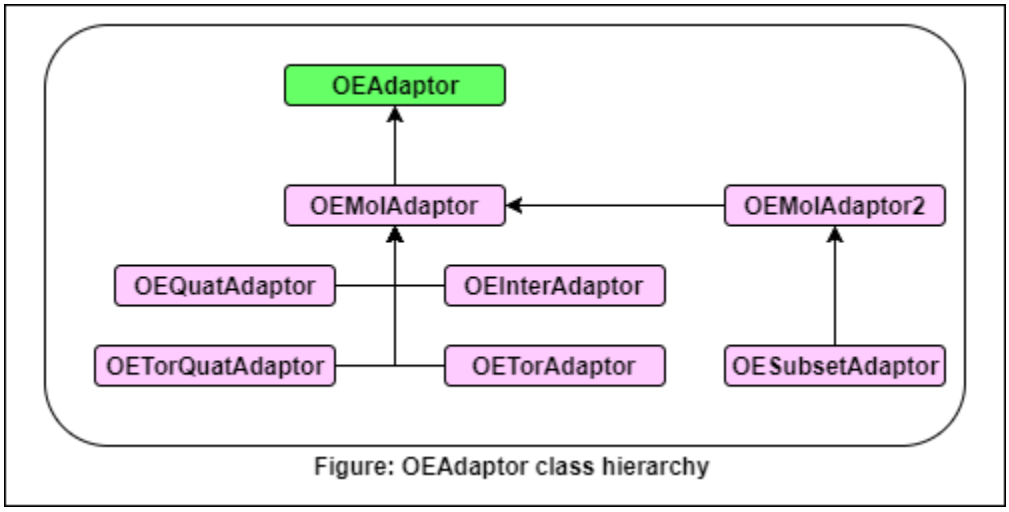


Figure: OEAdaptor class hierarchy

The *OEAdaptor* class defines the following public methods:

- *AdaptVar*
- *GetVar*

The following classes derive from this class:

- *OEInterAdaptor*

- *OEMolAdaptor*
- *OEMolAdaptor2*
- *OEQuatAdaptor*
- *OESubsetAdaptor*
- *OETorAdaptor*
- *OETorQuatAdaptor*

AdaptVar

```
bool AdaptVar(double* ca, const double *) const
```

Takes a set of variables as the second argument and copies a set of transformed variables into the first argument. This method is commonly used to transform optimized variables back into the coordinate system of the original reference variables.

GetVar

```
void GetVar(double* var, const double* refVar)
```

Takes reference variables as an argument (`refVar`), and transforms them into the variable type or coordinate system (`var`) that will be used during an optimization. Reference variables normally use the same coordinate system as the function evaluations that will be performed in an optimization. Many adaptors take Cartesian coordinates as the argument to the method, and return an array of variables containing the initial adapted state, such as torsion angles or a quaternion.

3.1.2 OEBFGSOpt

> **Attention**
>
> This API is currently available in C++ and Python.

```
class OEBFGSOpt : public OEOptimizer1
```

The *OEBFGSOpt* implements the quasi-Newton optimization method with updating Hessian according to the scheme of Broyden, Fletcher, Goldfarb and Shanno. The convergence criteria in based on $\sum_i g_i g_i$.

The following methods are publicly inherited from *OEOptimizer1*:

- *operator()*
- *GetIterLimit* and *SetIterLimit*
- *SetLineMinimize*
- *GetTolerance* and *SetTolerance*

The *OEBFGSOpt* class defines the following public methods:

- *GetHessianDimension*
- *GetInverseHessian*
- *GetKeepHessian*
- *SetKeepHessian*

- *UseSavedHessian*

Constructors

```
OEBFGSOpt()
OEBFGSOpt(const OEBFGSOpt &)
```

Default and copy constructors.

GetHessianDimension

```
unsigned int GetHessianDimension() const
```

Returns the dimension of the Hessian matrix.

GetInverseHessian

```
bool GetInverseHessian(double* h, unsigned int nv) const
```

Copies the inverted Hessian matrix into the array `h` in the form of lower triangle of complete Hessian. This function is useful when the Hessian matrix generated during optimization is needed. `nv` is the number of optimized variables.

GetKeepHessian

```
bool GetKeepHessian() const
```

Returns `true` if the *OEBFGSOpt* object is set to keep its Hessian

SetKeepHessian

```
void SetKeepHessian(bool keep)
```

If the value of the argument passed is `true`, the *OEBFGSOpt* object will keep its Hessian when the optimization is done.

UseSavedHessian

```
void UseSavedHessian(bool ini)
```

By default, at the start of the optimization Hessian is initiated as unit matrix. In the case when optimization has to repeated, for example starting with different values of variables, one can use this method to initiate optimization with the Hessian from the previous run.

3.1.3 OECGLineMinimize

> **Attention**
>
> This API is currently available in C++ and Python.

```
class OECGLineMinimize : public OELineMinimize
```

The *OECGLineMinimize* class implements line minimization used by the Conjugate Gradient optimizer *OEConjGradOpt*

The following methods are publicly inherited from *OELineMinimize*:

- *operator()*
- *CreateCopy*
- *SetMaxStep*

Constructors

```
OECGLineMinimize()
```

Default and copy constructors.

3.1.4 OECheckpoint0

> **Attention**
>
> This API is currently available in C++ and Python.

```
class OECheckpoint0
```

The *OECheckpoint0* class is an abstract class which provides a "callback" facility to monitor the progress of an optimization. In addition, the checkpoint class can also be used to control the termination of an optimization. This class is specifically for use with optimizers which do not use gradients.

The *OECheckpoint0* class defines the following public methods:

- *operator()*

The following classes derive from this class:

- *OECheckpoint1*

operator()

```
bool operator()(unsigned int iteration, unsigned int nvar, double fval, const double*␣
↪var = (double*)0, unsigned int state = 0) = 0
```

This operator method is called by an optimizer at least once per iteration during an optimization. The variables passed to the method from the optimizer reflect the current state of an optimization.

iteration
: The iteration number.

nvar
: The number of variables.

fval
: The most recent function evaluation during the current iteration.

var
: Array containing current set of optimized variables. If available, the array has a length of `nvar`.

state
: The current state of optimization.

3.1.5 OECheckpoint1

> **Attention**
>
> This API is currently available in C++ and Python.

```
class OECheckpoint1 : public OECheckpoint0
```

The *OECheckpoint1* provides a "callback" facility to monitor the progress of an optimization. In addition, it can also be used to control the termination of an optimization. This class may be used with optimizers that do not use gradients as well as those that use gradients

The following methods are publicly inherited from *OECheckpoint0*:

- *operator()*

The *OECheckpoint1* class defines the following public methods:

- *operator()*

operator()

```
bool operator()(unsigned int iteration, unsigned int nvar, double fval, double gnorm,␣
↪const double* var = (double *) 0, const double* grad = (double *) 0, unsigned int␣
↪state = 0)=0
```

This operator method is called by an optimizer at least once per iteration during an optimization. The variables passed to the method from the optimizer reflect the current state of an optimization.

iteration
: The iteration number.

nvar
: The number of variables.

fval
: The most recent function evaluation during the current iteration.

gnorm
: The most recent gradient norm evaluation during the current iteration. The gradient norm is defined as $\sqrt{\sum_i g_i g_i}$

var
: Array containing current set of optimized variables. If available, the array has a length of `nvar`.

grad
: Array containing current set of optimized variables. If available, the array has a length of `nvar`.

state
: The current state of optimization.

3.1.6 OEConjGradOpt

> **Attention**
>
> This API is currently available in C++ and Python.

```
class OEConjGradOpt : public OEOptimizer1
```

The *OEConjGradOpt* implements the conjugate gradient optimization method. The convergence criteria in based on $\sum_i g_i g_i$.

The following methods are publicly inherited from *OEOptimizer1*:

- *operator()*
- *GetIterLimit* and *SetIterLimit*
- *SetLineMinimize*
- *GetTolerance* and *SetTolerance*

Constructors

```
OEConjGradOpt()
OEConjGradOpt(const OEConjGradOpt &)
```

Default and copy constructors.

3.1.7 OEDFPOpt

> **Attention**
>
> This API is currently available in C++ and Python.

```
class OEDFPOpt : public OEOptimizer1
```

The *OEDFPOpt* implements the quasi-Newton optimization according to the Davidon-Fletcher-Powell method. The convergence criteria in based on $\sum_i g_i g_i$.

The following methods are publicly inherited from *OEOptimizer1*:

- *operator()*
- *GetIterLimit* and *SetIterLimit*
- *SetLineMinimize*
- *GetTolerance* and *SetTolerance*

Constructors

```
OEDFPOpt()
OEDFPOpt(const OEDFPOpt &)
```

Default and copy constructors.

3.1.8 OEFComponent

> **Attention**
>
> This API is currently available in C++ and Python.

```
struct OEFComponent
```

The *OEFComponent* class associates interaction-component names with with computed energy values. Methods which override `GetFComponents` return an iterator over *OEFComponent* instances.

Constructors

```
OEFComponent()
OEFComponent(const char* n, double v)
```

Default and copy constructors.

Second constructor makes an instance of *OEFComponent* given a name and value for an interaction.

3.1.9 OEFletcherReevesOpt

> **Attention**
>
> This API is currently available in C++ and Python.

```
class OEFletcherReevesOpt : public OEOptimizer1
```

The *OEFletcherReevesOpt* class implements the Fletcher-Reeves optimization method belonging to the family of conjugate gradient methods. The convergence criteria in based on $\sum_i g_i g_i$.

The following methods are publicly inherited from *OEOptimizer1*:

- *operator()*
- *GetIterLimit* and *SetIterLimit*
- *SetLineMinimize*
- *GetTolerance* and *SetTolerance*

Constructors

```
OEFletcherReevesOpt()
OEFletcherReevesOpt(const OEFletcherReevesOpt &)
```

Default and copy constructors.

3.1.10 OEFunc0

Attention

This API is currently available in C++ and Python.

class OEFunc0

The *OEFunc0* is an abstract base class. This class defines the interface for functions which take a set of variables and compute a function value, but have no defined analytical gradients. Classes derived from *OEFunc0* are intended for use with optimizers which do not require gradients.

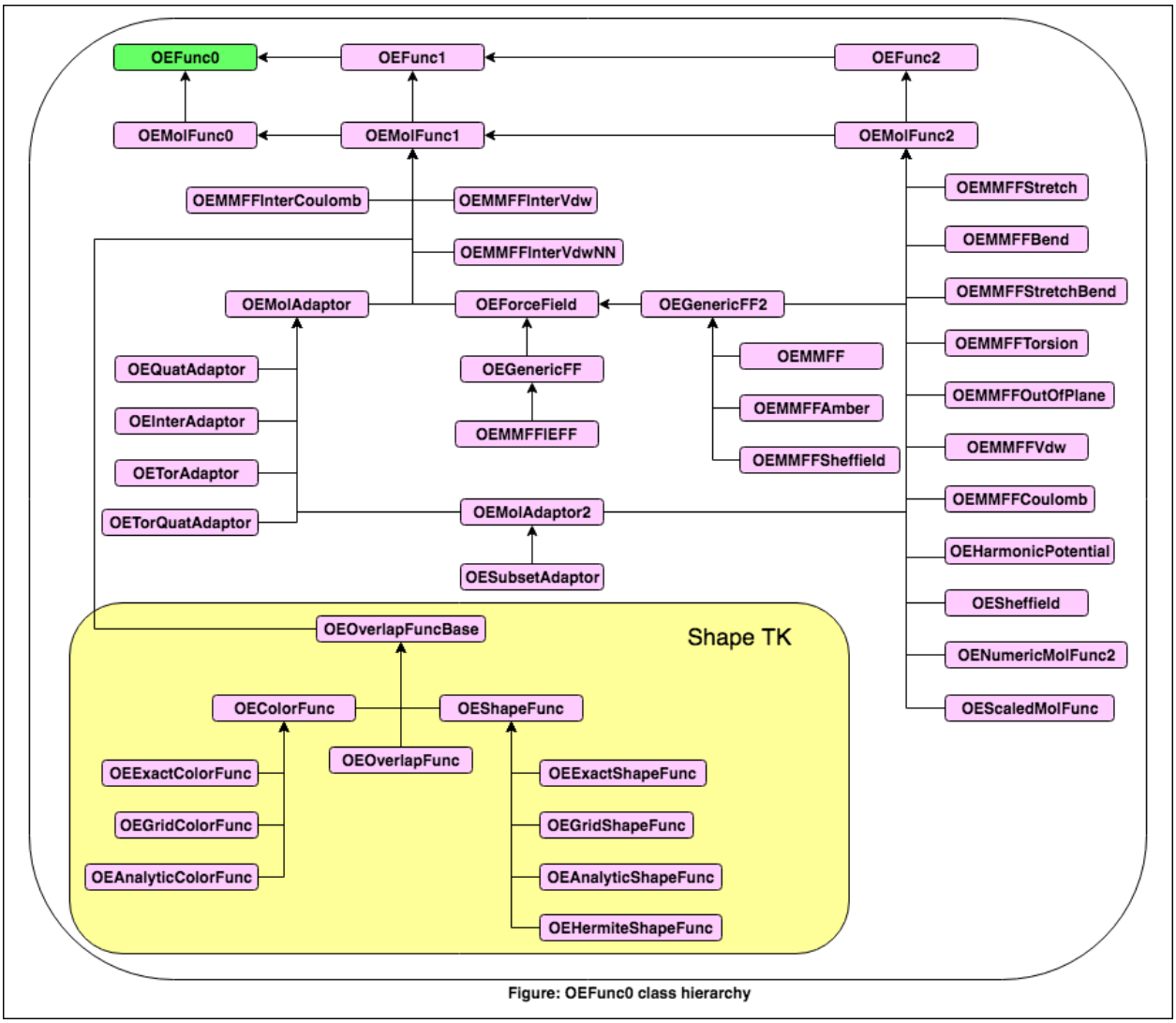


Figure: OEFunc0 class hierarchy

The *OEFunc0* class defines the following public methods:

- *operator()*
- *GetFComponents*
- *NumVar*

The following classes derive from this class:

- *OEFunc1*
- *OEFunc2*
- *OEMolFunc0*
- *OEMolFunc1*
- *OEMolFunc2*
- *OEForceField*
- *OEGenericFF*
- *OEGenericFF2*
- *OEInterAdaptor*
- *OEMolAdaptor*
- *OEQuatAdaptor*
- *OESubsetAdaptor*
- *OETorAdaptor*
- *OETorQuatAdaptor*
- *OENumericMolFunc2*
- *OEScaledMolFunc*
- *OEFFPotential*
- *OEStretchPotential*
- *OEBendPotential*
- *OETorsionPotential*
- *OEOutOfPlanePotential*
- *OENonBondPotBase*
- *OEVdwPotential*
- *OECoulombPotential*
- *OEInterMolFunc1*
- *OEInterMolFunc2*
- *OEInterNonBondPotBase*
- *OEInterNonBondPotential*
- *OEInterVdwPotential*
- *OEInterCoulombPotential*
- *OEHarmonicPotential*
- *OEMMFF*
- *OEMMFFStretchBend*
- *OEMMFFIEFF*
- *OESheffield*
- OEOverlapFuncBase

- OEOverlapFunc
- OEShapeFunc
- OEExactShapeFunc
- OEAnalyticShapeFunc
- OEHermiteShapeFunc
- OEGridShapeFunc
- OEColorFunc
- OEExactColorFunc
- OEAnalyticColorFunc
- OEGridColorFunc

operator()

```
double operator()(const double*)
```

This method defines the interface for function evaluation. An array of variables is passed as an argument, and a function value is returned. All classes derived from *OEFunc0* implements this method.

GetFComponents

```
OESystem::OEIterBase<OEFComponent>* GetFComponents(const double *)
```

This method defines the interface for retrieving information about function names and associated function values, given a set of variables. The method takes an array of variables, and returns an iterator over the names and component values (typically energies).

All OE classes derived from *OEFunc0*, implements this method. However, due to the unique construct of the return value from this method, it is not possible to implement this in a user derived class. It is suggested to return a NULL for user derived classes and handle things accordingly.

NumVar

```
unsigned int NumVar() const
```

This method defines the interface for returning the number of function variables. All classes derived from *OEFunc0* implements this method.

3.1.11 OEFunc1

> **Attention**
>
> This API is currently available in C++ and Python.

```
class OEFunc1 : public OEFunc0
```

The *OEFunc1* is an abstract base class. This class defines the interface for functions which take a set of variables and compute a function value and gradients. Classes derived from *OEFunc1* are intended for use with optimizers which require gradients.

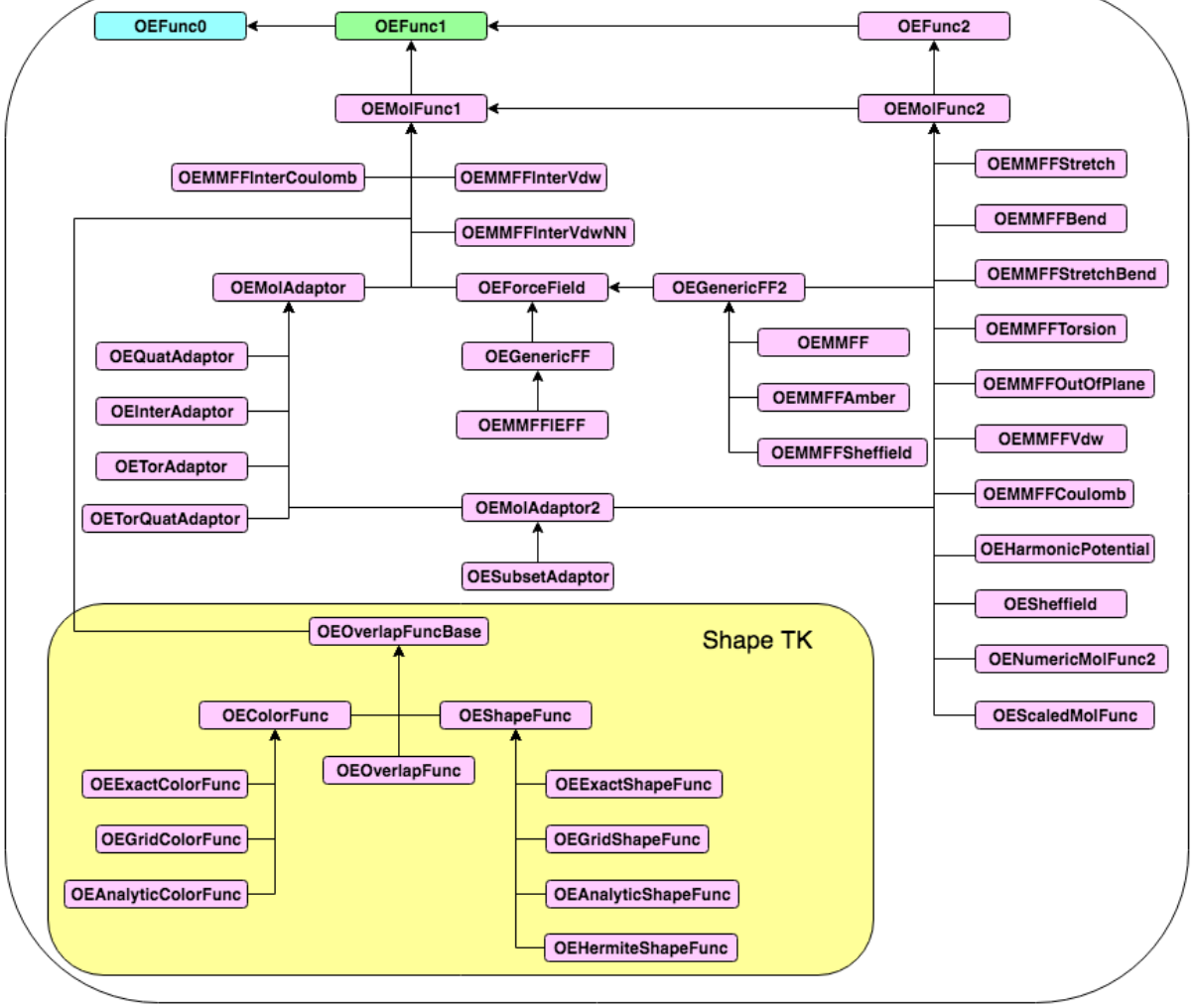


Figure: OEFunc1 class hierarchy

The following methods are publicly inherited from *OEFunc0*:

- *operator()*
- *GetFComponents*
- *NumVar*

The *OEFunc1* class defines the following additional public methods:

- *operator()*

The following classes derive from this class:

- *OEFunc2*
- *OEMolFunc1*
- *OEMolFunc2*
- *OEForceField*
- *OEGenericFF*
- *OEGenericFF2*

- *OEInterAdaptor*
- *OEMolAdaptor*
- *OEQuatAdaptor*
- *OESubsetAdaptor*
- *OETorAdaptor*
- *OETorQuatAdaptor*
- *OENumericMolFunc2*
- *OEScaledMolFunc*
- *OEFFPotential*
- *OEStretchPotential*
- *OEBendPotential*
- *OETorsionPotential*
- *OEOutOfPlanePotential*
- *OENonBondPotBase*
- *OEVdwPotential*
- *OECoulombPotential*
- *OEInterMolFunc1*
- *OEInterMolFunc2*
- *OEInterNonBondPotBase*
- *OEInterNonBondPotential*
- *OEInterVdwPotential*
- *OEInterCoulombPotential*
- *OEHarmonicPotential*
- *OEMMFF*
- *OEMMFFStretchBend*
- *OEMMFFIEFF*
- *OESheffield*
- OEOverlapFuncBase
- OEOverlapFunc
- OEShapeFunc
- OEExactShapeFunc
- OEAnalyticShapeFunc
- OEHermiteShapeFunc
- OEGridShapeFunc
- OEColorFunc
- OEExactColorFunc

- OEAnalyticColorFunc
- OEGridColorFunc

operator()

```
double operator()(const double* x, double* g)
```

This method defines the interface for function evaluation along with the gradients. An array of variables is passed as an argument, and a function value is returned. The corresponding gradients for the given set of variable is returned in the second argument. All classes derived from *OEFunc1* implements this method.

There should be a one to one correspondence between the elements of the variable and gradient arrays. Methods that override this operator method must not initialize the gradient array, but rather assume the array has already been initialized.

3.1.12 OEFunc2

> **Attention**
>
> This API is currently available in C++ and Python.

```
class OEFunc2 : public OEFunc1
```

The *OEFunc2* is an abstract base class. This class defines the interface for functions which take a set of variables and compute a function value, gradients and second derivatives.

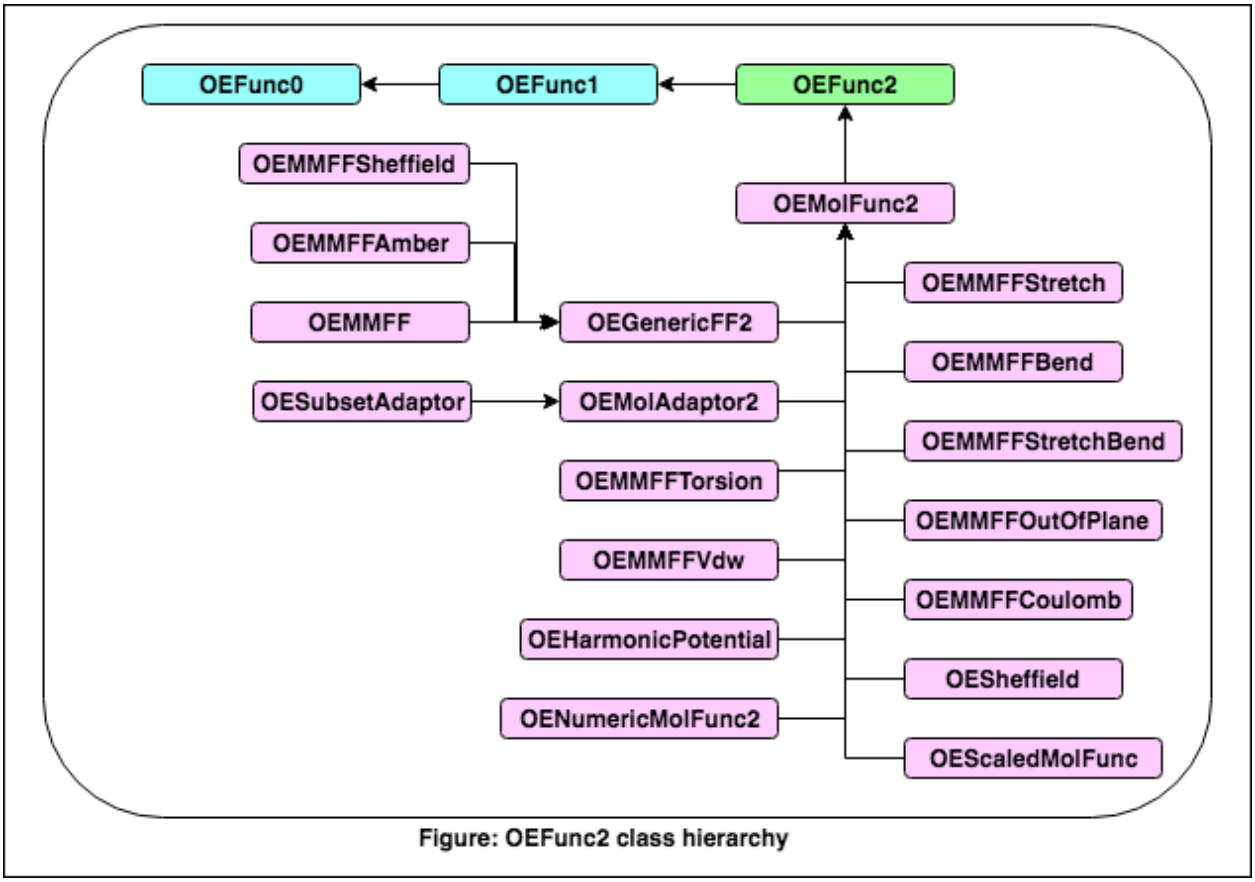


Figure: OEFunc2 class hierarchy

The following methods are publicly inherited from *OEFunc0*:

- *operator()*
- *GetFComponents*
- *NumVar*

The following methods are publicly inherited from *OEFunc1*:

- *operator()*

The *OEFunc2* class defines the following additional public methods:

- *operator()*

The following classes derive from this class:

- *OEMolFunc2*
- *OEGenericFF2*
- *OESubsetAdaptor*
- *OENumericMolFunc2*
- *OEScaledMolFunc*
- *OEFFPotential*
- *OEStretchPotential*
- *OEBendPotential*

- *OETorsionPotential*
- *OEOutOfPlanePotential*
- *OENonBondPotBase*
- *OEVdwPotential*
- *OECoulombPotential*
- *OEInterMolFunc2*
- *OEInterNonBondPotBase*
- *OEInterNonBondPotential*
- *OEInterVdwPotential*
- *OEInterCoulombPotential*
- *OEMMFF*
- *OEMMFFStretchBend*
- *OESheffield*
- OEHermiteShapeFunc

operator()

```
bool operator()(const double* x, double* h, double* g)
```

This method defines the interface for function evaluation along with gradients and second derivatives. The corresponding second derivatives for the given set of variable is returned in the second argument, and the gradients are returned in the third argument.

There should be a one to one correspondence between the elements of the variable and that second derivatives and gradients arrays. Methods that override this operator method must not initialize the second derivatives and the gradient arrays, but rather assume that the arrays has already been initialized.

3.1.13 OEHestenesStiefelOpt

> **Attention**
>
> This API is currently available in C++ and Python.

```
class OEHestenesStiefelOpt : public OEOptimizer1
```

The *OEHestenesStiefelOpt* class implements the Hestenes-Stiefel optimization method belonging to the family of conjugate gradient methods. The convergence criteria in based on $\sum_i g_i g_i$.

The following methods are publicly inherited from *OEOptimizer1*:

- *operator()*
- *GetIterLimit* and *SetIterLimit*
- *SetLineMinimize*
- *GetTolerance* and *SetTolerance*

Constructors

```
OEHestenesStiefelOpt()
OEHestenesStiefelOpt(const OEHestenesStiefelOpt& rhs)
```

Default and copy constructors.

3.1.14 OELineMinimize

> **Attention**
>
> This API is currently available in C++ and Python.

```
class OELineMinimize
```

The *OELineMinimize* class is a base class which defines the interface for line minimization used in different types of optimization.

The *OELineMinimize* class defines the following public methods:

- *operator()*
- *CreateCopy*
- *SetMaxStep*

The following classes derive from this class:

- *OECGLineMinimize*
- *OEMaxStepLineMinimize*
- *OEQNLineMinimize*
- *OENRLineSearch*

operator()

```
bool operator()(unsigned int n, const double* p, double* d, double* g, double* k,␣
↪double* gmin, double* alpha, OEFunc1& fun ) const = 0
```

Defines the interface for finding the step size `alpha` for the current optimizer iteration in the search direction determined by the vector `d`. `n` is the number of optimized variables, array `p` and `g` contain initial coordinates and gradients, while array `k` returns gradients for the coordinates determined by the returned step size `alpha`. `gmin` contains the initial function value and returns its optimized value, and `fun` is a reference to optimized function object. Operator returns `true` when a value lower than the initial function value is found, `false` otherwise.

SetMaxStep

```
bool SetMaxStep(unsigned int , double *)
```

Defines the interface for determination of the maximum step size. This method has a trivial implementation in the base class, and provides a way to determine the maximum step if the specific line optimization derived from *OELineMinimize* needs it.

CreateCopy

```
OELineMinimize* CreateCopy() const
```

Defines the interface for creating a copy of line minimization object.

3.1.15 OEMaxStepLineMinimize

> **Attention**
>
> This API is currently available in C++ and Python.

```
class OEMaxStepLineMinimize : public OELineMinimize
```

The following methods are publicly inherited from *OELineMinimize*:

- *operator()*
- *CreateCopy*
- *SetMaxStep*

Constructors

```
OEMaxStepLineMinimize()
OEMaxStepLineMinimize(double* maxStep, unsigned len)
OEMaxStepLineMinimize(const OEMaxStepLineMinimize& rhs)
```

Default and copy constructors.

3.1.16 OENRLineSearch

> **Attention**
>
> This API is currently available in C++ and Python.

```
class OENRLineSearch : public OELineMinimize
```

The *OENRLineSearch* class implements line minimization using backtracking line search method.

The following methods are publicly inherited from *OELineMinimize*:

- *operator()*
- *SetMaxStep*
- *CreateCopy*

Constructors

```
OENRLineSearch(unsigned int inistep = OENROpts::UseFullNewtonInitialStep)
```

Default and copy constructors.

3.1.17 OEOptimizer0

> **Attention**
>
> This API is currently available in C++ and Python.

```
class OEOptimizer0
```

The *OEOptimizer0* abstract class defines the interface for optimizing a set of variables for which a function can be evaluated. Implementations of *OEOptimizer0* use only function values during optimization, so it does not calculate gradients.

The *OEOptimizer0* class defines the following public methods:

- *operator()*
- *GetIterLimit* and *SetIterLimit*

operator()

```
double operator()(OEFunc0& function, const double* var, double* opt_var)
double operator()(OEFunc0& function, OECheckpoint0* check, const double* var, double*␣
↪opt_var)
```

These methods define the interface for optimizing a set of variables with the use of evaluated function values. The reference to the function object is passed as a first argument. Initial values of variables are passed in the array `var`. The optimized variables are returned in the array `opt_var`. Both operators return the value of the corresponding optimized function value. The second operator takes a pointer to the *OECheckpoint0* object that can be used to monitor the progress of an optimization.

GetIterLimit

```
unsigned GetIterLimit() const
```

See *SetIterLimit*.

SetIterLimit

```
void SetIterLimit(unsigned int itmax)
```

Defines the interface for setting the maximum number of iterations that an optimizer derived from the *OEOptimizer0* will attempt while trying to identify a converged minimum function value. Optimization will cease if the iteration limit is reached without finding a converged minimum.

3.1.18 OEOptimizer1

> **Attention**
>
> This API is currently available in C++ and Python.

```
class OEOptimizer1
```

The *OEOptimizer1* abstract class defines the interface for optimizing a set of variables for which a function and gradients can be evaluated. Implementations of *OEOptimizer1* use both function values and gradients during optimization.

The *OEOptimizer1* class defines the following public methods:

- *operator()*
- *GetIterLimit* and *SetIterLimit*
- *SetLineMinimize*
- *GetTolerance* and *SetTolerance*

The following classes derive from this class:

- *OEBFGSOpt*
- *OEConjGradOpt*
- *OEDFPOpt*
- *OEFletcherReevesOpt*
- *OEHestenesStiefelOpt*
- *OERibierePolakOpt*
- *OESteepestDescentOpt*

operator()

```
double operator()(OEFunc1& function, const double* var, double* opt_var)
double operator()(OEFunc1& function, OECheckpoint1* check, const double* var, double*␣
↪opt_var)
```

These virtual methods define the interface for optimizing a set of variables with the use of evaluated function values. The reference to the function object is passed as a first argument. Initial values of variables are passed in the array `var`. The optimized variables are returned in the array `opt_var`. Both operators return the value of the corresponding optimized function value. The second operator takes a pointer to the *OECheckpoint1* object that can be used to monitor the progress of an optimization.

GetIterLimit

```
unsigned GetIterLimit() const
```

See *SetIterLimit*.

GetTolerance

```
double GetTolerance() const
```

See *SetTolerance*.

SetIterLimit

```
void SetIterLimit(unsigned int itmax)
```

Defines the interface for setting the maximum number of iterations that an optimizer derived from the *OEOptimizer0* will attempt while trying to identify a converged minimum function value. Optimization will cease if the iteration limit is reached without finding a converged minimum.

SetLineMinimize

```
bool SetLineMinimize(const OELineMinimize &linmin)
```

Allows selection of the line minimization object used by the optimizer.

SetTolerance

```
void SetTolerance(double t)
```

Defines the interface for setting the gradient convergence criteria. Depending on the implementation it might be the root mean square gradient, gradient norm or gradient dot product. Optimization will terminate normally if the convergence criteria set by this method is achieved during an optimization.

3.1.19 OEOptimizer2

> **Attention**
>
> This API is currently available in C++ and Python.

```
class OEOptimizer2
```

The *OEOptimizer2* abstract class defines the interface for optimizing a set of variables for which a function, gradients and second derivatives (elements oh Hessian matrix) can be evaluated. Implementations of *OEOptimizer2* use function values, gradients and second derivatives of function during optimization.

The *OEOptimizer2* class defines the following public methods:

- *operator()*
- *GetIterLimit* and *SetIterLimit*
- *GetTolerance* and *SetTolerance*

The following classes derive from this class:

- *OENewtonOpt*

operator()

```
double operator()(OEFunc2& function, const double* var, double* opt_var)
double operator()(OEFunc2& function, OECheckpoint1* check, const double* var, double*␣
↪opt_var)
```

These virtual methods define the interface for optimizing a set of variables with the use of evaluated function values. The reference to the function object is passed as a first argument. Initial values of variables are passed in the array `var`. The optimized variables are returned in the array `opt_var`. Both operators return the value of the corresponding optimized function value. The second operator takes a pointer to the *OECheckpoint1* object that can be used to monitor the progress of an optimization.

GetIterLimit

```
unsigned GetIterLimit() const
```

See *SetIterLimit*.

GetTolerance

```
double GetTolerance() const
```

See *SetTolerance*.

SetIterLimit

```
void SetIterLimit(unsigned int itmax)
```

Defines the interface for setting the maximum number of iterations that an optimizer derived from the *OEOptimizer2* will attempt while trying to identify a converged minimum function value. Optimization will cease if the iteration limit is reached without finding a converged minimum.

SetTolerance

```
void SetTolerance(double t)
```

Defines the interface for setting the gradient convergence criteria. Depending on the implementation it might be the root mean square gradient, gradient norm or gradient dot product. Optimization will terminate normally if the convergence criteria set by this method is achieved during an optimization.

3.1.20 OEQNLineMinimize

> **Attention**
>
> This API is currently available in C++ and Python.

```
class OEQNLineMinimize : public OELineMinimize
```

The *OEQNLineMinimize* class implements line minimization for the optional use in quasi-Newton optimization.

The following methods are publicly inherited from *OELineMinimize*:

- *operator()*
- *CreateCopy*
- *SetMaxStep*

Constructors

```
OEQNLineMinimize(double alpha = 1e-4, double beta = 0.9,
                 double approxzero = 1e-10, unsigned int lmmax = 32)
```

Default and copy constructors.

3.1.21 OERibierePolakOpt

> **Attention**
>
> This API is currently available in C++ and Python.

```
class OERibierePolakOpt : public OEOptimizer1
```

The *OERibierePolakOpt* class implements the Polak-Ribière optimization method belonging to the family of conjugate gradient methods. The convergence criteria in based on $\sum_i g_i g_i$.

The following methods are publicly inherited from *OEOptimizer1*:

- *operator()*
- *GetIterLimit* and *SetIterLimit*
- *SetLineMinimize*
- *GetTolerance* and *SetTolerance*

Constructors

```
OERibierePolakOpt()
OERibierePolakOpt(const OERibierePolakOpt& rhs)
```

Default and copy constructors.

3.1.22 OESteepestDescentOpt

> **Attention**
>
> This API is currently available in C++ and Python.

```
class OESteepestDescentOpt : public OEOptimizer1
```

The *OESteepestDescentOpt* class implements the steepest descent optimization method. The convergence criteria in based on $\sum_i g_i g_i$.

The following methods are publicly inherited from *OEOptimizer1*:

- *operator()*
- *GetIterLimit* and *SetIterLimit*
- *GetTolerance* and *SetTolerance*

The *OESteepestDescentOpt* class defines the following public methods:

- *SetInitialStep*

Constructors

```
OESteepestDescentOpt()
```

Default and copy constructors.

SetInitialStep

```
void SetInitialStep(double step)
```

Sets the initial step. The default initial step is $\sqrt{(1.0/\sum_i g_i g_i)}$

3.1.23 OENewtonOpt

> **Attention**
>
> This API is currently available in C++ and Python.

```
class OENewtonOpt : public OEOptimizer2
```

The *OENewtonOpt* implements the Newton-Raphson optimization method using analytical Hessian and its Cholesky decomposition for determination of current search direction.

The following methods are publicly inherited from *OEOptimizer2*:

- *operator()*
- *GetIterLimit* and *SetIterLimit*
- *GetTolerance* and *SetTolerance*

Constructors

```
OENewtonOpt()
OENewtonOpt(const OENewtonOpt &)
```

Default and copy constructors.

> **See also**
>
> - *Solving a simple equation* example

Assignment operator.

3.2 OEMolPotential Classes

3.2.1 OEBendParams

> **Attention**
>
> This API is currently available in C++ and Python.

```
class  OEBendParams : public OEMolPotential::OEInteractParams
```

The *OEBendParams* class defines generic angle-bend parameters used in any force field which has bond angle term.

The following methods are publicly inherited from *OEInteractParams*:

- *Clear*
- *Get*
- *Set*

The *OEBendParams* class defines the following public methods:

- *GetAtomIndex1*
- *GetAtomIndex2*
- *GetAtomIndex3*
- *IsValid*
- *IsAngle*

Constructors

```
OEBendParams(const OEChem::OEBondBase* bond1,
             const OEChem::OEBondBase* bond2,
             const unsigned type = OEFormatType::AMBER)
OEBendParams(const OEBendParams&)
```

Constructor and copy constructor. Constructs an *OEBendParams* object for two bonds which make the bond angle.

GetAtomIndex1

```
unsigned GetAtomIndex1() const
```

Returns atom index of the first atom for bond angle for which the instance of *OEBendParams* was created.

GetAtomIndex2

```
unsigned GetAtomIndex2() const
```

Returns atom index of the second atom for bond angle for which the instance of *OEBendParams* was created.

GetAtomIndex3

```
unsigned GetAtomIndex3() const
```

Returns atom index of the third atom for bond angle for which the instance of *OEBendParams* was created.

IsValid

```
bool IsValid() const
bool IsValid(const OESystem::OEUnaryPredicate<OEChem::OEAtomBase>& pred) const
bool IsValid(const OESystem::OEBinaryPredicate<OEChem::OEBondBase, OEChem::OEBondBase>&␣
↪pred) const
```

Methods which check if an instance of the *OEBendParams* class was created for the bond angle to be included in force field calculation. First method is checking if the above instance was correctly constructed, while two remaining methods provide predicates to test if atoms (second method) and two bonds which define bond angle belong to that instance.

IsAngle

```
bool IsAngle(const OEChem::OEAtomBase* atom1,
             const OEChem::OEAtomBase* atom2,
             const OEChem::OEAtomBase* atom3) const
```

Used to check if 3 atom passed as an arguments belong to the current *OEBendParams* instance.

3.2.2 OEBendPotential

> **Attention**
>
> This API is currently available in C++ and Python.

```
class OEBendPotential : public virtual OEMolPotential::OEFFPotential
```

The *OEBendPotential* defines an interface all angle bending force field interaction potentials.

The following methods are publicly inherited from *OEFunc0*:

- *operator()*
- *GetFComponents*
- *NumVar*

The following methods are publicly inherited from *OEFunc1*:

- *operator()*

The following methods are publicly inherited from *OEFunc2*:

- *operator()*

The following methods are publicly inherited from *OEMolFunc*:

- *Clear*
- *ClearPredicates*
- *GetInteractions*
- *Setup*

The following methods are publicly inherited from *OEFFPotential*:

- *Set*

The *OEBendPotential* implements the following methods:

- *Set*

Constructors

```
OEBendPotential(const OEFFParams&)
OEBendPotential(const OEBendPotential&)
```

Constructor and copy constructor.

Assignment operator

```
OEBendPotential& operator=(const OEBendPotential&)
```

Assignment operator.

Set

```
bool Set(const OESystem::OEBinaryPredicate<OEChem::OEBondBase, OEChem::OEBondBase>&,
         unsigned int flag = OEFuncType::Bend) override
```

This method allows interaction-level control of the angle-bend term. The predicate passed is used to test the molecule's bond angles. Only those bond angles which are reported as `true` are included in the list of interactions to be calculated. The *Set* method must be called before the *Setup* for it to be effective.

3.2.3 OECoulombPotential

> **Attention**
>
> This API is currently available in C++ and Python.

```
class OECoulombPotential : public virtual OENonBondPotBase
```

The *OECoulombPotential* class defines an interface for calculation of intra-molecular Coulomb force field interaction potentials.

The following methods are publicly inherited from *OEFunc0*:

- *operator()*
- *GetFComponents*
- *NumVar*

The following methods are publicly inherited from *OEFunc1*:

- *operator()*

The following methods are publicly inherited from *OEFunc2*:

- *operator()*

The following methods are publicly inherited from *OEMolFunc*:

- *Clear*
- *ClearPredicates*
- *GetInteractions*
- *Setup*

The following methods are publicly inherited from *OEFFPotential*:

- *Set*

The following methods are publicly inherited from *OENonBondPotBase*:

- *GetNonBondOptions*
- *Set*

Constructors

```
OECoulombPotential(const OEMolPotential::OEFFParams& fp,
                   const double rcut = 0.0);
OECoulombPotential(const OECoulombPotential&)
```

Constructor and copy constructor.

Assignment operator

```
OECoulombPotential& operator=(const OECoulombPotential&)
```

Assignment operator.

3.2.4 OEFFParams

> **Attention**
>
> This API is currently available in C++ and Python.

```
class OEFFParams
```

The *OEFFParams* is an abstract base class. This class defines the interface to retrieve parameters defined by a force field. In addition, simple methods are included for facile extraction of idealized geometric information base on the force field.

The *OEFFParams* class defines the following public methods:

- *CreateCopy*
- *GetAtomTypeIndex*
- *GetAtomTypeName*
- *GetContactDistance*
- *GetCoulombParams*
- *GetEquilibriumAngle*
- *GetEquilibriumBondLength*
- *GetStretchBendParams*
- *GetVdwRadius*
- *PrepMol*
- *GetVdwParams*
- *GetStretchParams*
- *GetBendParams*
- *GetTorsionParams*
- *GetOutOfPlaneParams*
- *HasVdwParams*
- *HasStretchParams*

- *HasBendParams*
- *HasTorsionParams*
- *GetTitle*

The following classes derive from this class:

- *OEMMFF94sParams*
- *OEMMFFParams*
- *OESmirnoffParams*

Constructors

```
OEFFParams()
OEFFParams(const OEFFParams&)
```

Constructor and copy constructor.

Assignment operator

```
OEFFParams& operator=(const OEFFParams&)
```

Assignment operator.

CreateCopy

```
OEFFParams* CreateCopy() const
```

This method returns a deep copy of the *OEFFParams* derived object. The copy of the object returned is in memory owned by the user making the function call. The delete operator must be called for the returned pointer when the copy is no longer needed in order to avoid memory leaks.

GetAtomTypeIndex

```
unsigned int GetAtomTypeIndex(unsigned int) const
```

Provides interface to get an atom type index for the atom identified by the force field type passed as an argument. The point is that a number of chemically different atom types in the force field can belong to a single type for which force field parameters are defined.

GetAtomTypeName

```
const char* GetAtomTypeName(unsigned int) const
```

Provides interface to convert the integer force field atom type into a symbolic type.

GetContactDistance

```
bool GetContactDistance(double &, const OEChem::OEAtomBase *,
                        const OEChem::OEAtomBase *) const
```

Provides the interface for retrieving the distance between two atoms corresponding to the minimum on their VdW potential curve. The double precision floating point reference passed as the first argument to the method is assigned to the contact distance for the two atoms if it is defined by the force field. The method returns `true` only if a valid force

field parameter for the interaction exists. *OEFFParams.PrepMol* must be called for the molecule prior to using this method.

GetCoulombParams

```
bool GetCoulombParams(OEInteractParams &,
                      const OEChem::OEAtomBase *,
                      const OEChem::OEAtomBase *) const
```

Defines the interface for retrieving the parameters used in calculating an electrostatic interaction for two atoms. Overriding methods return `true` only if valid parameters exist for the atoms passed as the second and third arguments, otherwise they return `false`. *OEFFParams.PrepMol* must be called for the molecule prior to using this method.

GetEquilibriumAngle

```
bool GetEquilibriumAngle(double &, const OEChem::OEAtomBase *,
                         const OEChem::OEAtomBase *,
                         const OEChem::OEAtomBase *) const
```

This method is used to retrieve the equilibrium bond angle for the three atoms passed to it. The second atom passed to the method (`atom2`) must be the central atom of the requested bond angle. The double precision floating point reference passed as the first argument to the method is assigned to the equilibrium bond angle if it is defined by the force field. The method returns `true` only if a valid force field parameter for the interaction exists. *OEFFParams.PrepMol* must be called for the molecule prior to using this method.

GetEquilibriumBondLength

```
bool GetEquilibriumBondLength(double &, const OEChem::OEBondBase *) const
```

This method is used to retrieve the equilibrium bond length for the bond passed as the second argument. The double precision floating point reference passed as the first argument to the method is assigned to the equilibrium bond length if it is defined by the force field. The method returns `true` only if a valid force field parameter for the bond exists. *OEFFParams.PrepMol* must be called for the molecule prior to using this method.

GetStretchBendParams

```
bool GetStretchBendParams(OEInteractParams &,
                          const OEChem::OEAtomBase *,
                          const OEChem::OEAtomBase *,
                          const OEChem::OEAtomBase *) const
```

Defines the interface for retrieving the parameters used in calculating a stretch-bend interaction. The central atom of the bond-angle must be passed as the third argument to the method. Overriding methods return `true` only if valid parameters exist for the atoms passed as arguments, otherwise they should return `false`. *OEFFParams.PrepMol* must be called for the molecule prior to using this method.

GetVdwRadius

```
bool GetVdwRadius(double &, const OEChem::OEAtomBase* atm) const
```

Defines the interface for retrieving the VdW radius for an atom. Overriding methods return `true` only if a VdW radius can be calculated or retrieved for an atom, `false` otherwise. *OEFFParams.PrepMol* must be called for the molecule prior to using this method.

PrepMol

```
bool PrepMol(OEChem::OEMolBase& mol, bool sweep = true,
             bool closestTypeAllowed = true) const
```

Defines the interface for preparing a molecule for force field parameter retrieval. Atom types, names, partial charges, and indices stored within the molecule may be modified by overriding methods. The second argument to the method specifies whether a call to `OEMolBase.Sweep` is allowed during the preparation. By default, sweeping a molecule is allowed by the method. A return value of `true` indicates a successful preparation, however, if the method returns `false` calculations with the resultant modified molecule should not be attempted. The return value of this method should always be checked before use with its associated force field and components.

closestTypeAllowed
If *true* then the MMFF typing method will substitute close atom types if exact atom types do not exist.

See also

`OEMMFFAtomType` function

GetVdwParams

```
bool GetVdwParams(OEMolPotential::OEInteractParams&,
                  const OEChem::OEAtomBase*,
                  const OEChem::OEAtomBase*) const
```

Fills the first argument, *OEInteractParams*, with the Vdw parameters for the two non-bonded atoms specified by the second and the third arguments, OEAtomBase. If parameters were not found, it returns `false`; otherwise, it returns `true`.

```
OESystem::OEIterBase<OEMolPotential::OEVdwParams>*
OESystem::OEIterBase<OEMolPotential::OEVdwParams>* GetVdwParams(const OEChem::OEMolBase&␣
↪mol,
                                                                const␣
↪OESystem::OEUnaryPredicate<OEChem::OEAtomBase>* pred = 0,
                                                                const double rcut = 0.0)␣
↪const
```

Takes a molecule object, `mol`, and returns an iterator to all *OEVdwParams* objects created for that molecule. If the pointer to atom predicate is passed as the second argument, all atoms engaged in vdw non-bonded interactions will be tested against with that predicate. In this case, the returned iterator will contain parameters for only those pairs of atoms that pass the test. Similarly, all *OEVdwParams* objects for pairs outside the cutoff radius, `rcut`, will be eliminated. An `rcut` value of `0.0` or less assumes that all interactions should be included without any cutoff.

```
OESystem::OEIterBase<OEMolPotential::OEVdwParams>*
OESystem::OEIterBase<OEMolPotential::OEVdwParams>* GetVdwParams(const OEChem::OEMolBase&␣
↪mol,
                                                                const OEChem::OEMolBase&␣
↪host,
                                                                const double rcut = 0.0)␣
↪const
```

Returns an iterator to all *OEVdwParams* objects created for host-ligand interactions specified by the ligand molecule, `mol`, and the host molecule, `host`. All *OEVdwParams* objects for pairs outside the cutoff radius, `rcut`, will be eliminated. An `rcut` value of `0.0` or less assumes that all interactions should be included without any cutoff.

GetStretchParams

```
bool GetStretchParams(OEMolPotential::OEInteractParams&,
                      const OEChem::OEBondBase*) const
```

Fills the first argument, *OEInteractParams*, with the bond stretch parameters for the bond, `OEBondBase`. If the parameters were not found, it returns `false`; otherwise, it returns `true`.

```
OESystem::OEIterBase<OEMolPotential::OEStretchParams>*
OESystem::OEIterBase<OEMolPotential::OEStretchParams>* GetStretchParams(const␣
↪OEChem::OEMolBase& mol,
                                                                        const␣
↪OESystem::OEUnaryPredicate<OEChem::OEAtomBase>* pred = 0) const
```

Takes a molecule object, `mol`, and returns an iterator to all *OEStretchParams* objects created for that molecule. If the pointer to an atom predicate is passed as the second argument, all atoms making bonds will be tested against with that predicate. In this case, the returned iterator will contain parameters for only those bonds made between the atoms that pass the test.

GetBendParams

```
bool GetBendParams(OEMolPotential::OEInteractParams&,
                   const OEChem::OEAtomBase*,
                   const OEChem::OEAtomBase*,
                   const OEChem::OEAtomBase*) const
```

Fills the first argument, *OEInteractParams*, with the bend parameters for a bond angle made by 3 atoms, OEAtomBase, passed as the second, third and the forth arguments. If the parameters were not found, it returns `false`; otherwise, it returns `true`.

```
OESystem::OEIterBase<OEMolPotential::OEBendParams>*
OESystem::OEIterBase<OEMolPotential::OEBendParams>* GetBendParams(const␣
↪OEChem::OEMolBase& mol,
                                                                  const␣
↪OESystem::OEUnaryPredicate<OEChem::OEAtomBase>* pred = 0) const
```

Takes a molecule object, `mol`, and returns an iterator to all *OEBendParams* objects created for that molecule. If the pointer to an atom predicate is passed as the second argument, all atoms making bond angles will be tested against that with predicate and the returned iterator will contain parameters for only those bond angles whose atom members pass the test.

GetTorsionParams

```
bool GetTorsionParams(OEMolPotential::OEInteractParams&,
                      const OEChem::OEAtomBase*,
                      const OEChem::OEAtomBase*,
                      const OEChem::OEAtomBase*,
                      const OEChem::OEAtomBase*) const
```

Fills the first argument, *OEInteractParams*, with the torsion parameters for a torsion made by 4 atoms, OEAtomBase, passed as the second, third, forth and the fifth arguments. If the parameters were not found, it returns `false`; otherwise, it returns `true`.

```
OESystem::OEIterBase<OEMolPotential::OETorsionParams>*
OESystem::OEIterBase<OEMolPotential::OETorsionParams>* GetTorsionParams(const␣
↪OEChem::OEMolBase& mol,
                                                                        const␣
↪OESystem::OEUnaryPredicate<OEChem::OEAtomBase>* pred = 0) const
```

Takes a molecule object, `mol`, and returns an iterator to all *OETorsionParams* objects created for that molecule. If the pointer to an atom predicate is passed as the second argument, all atoms making torsions will be tested against with that predicate and the returned iterator will contain parameters for only those torsions whose atom members pass the test.

GetOutOfPlaneParams

```
bool GetOutOfPlaneParams(OEMolPotential::OEInteractParams&,
                         const OEChem::OEAtomBase*,
                         const OEChem::OEAtomBase*,
                         const OEChem::OEAtomBase*,
                         const OEChem::OEAtomBase*) const
```

Fills the first argument, *OEInteractParams*, with the improper torsion parameters for an improper torsion made by 4 atoms, OEAtomBase, passed as the second, third, forth and the fifth arguments. If the parameters were not found, it returns `false`; otherwise, it returns `true`.

```
OESystem::OEIterBase<OEMolPotential::OEOutOfPlaneParams>* GetOutOfPlaneParams(const␣
↪OEChem::OEMolBase& mol,
                                                                          const␣
↪OESystem::OEUnaryPredicate<OEChem::OEAtomBase>* pred = 0) const
```

Takes a molecule object, `mol`, and returns an iterator to all *OESmirnoffOutOfPlane* objects created for that molecule. If the pointer to an atom predicate is passed as the second argument, all atoms making improper torsions will be tested against with that predicate and the returned iterator will contain parameters for only those improper torsions whose atom members pass the test.

HasVdwParams

```
bool HasVdwParams(const OEChem::OEMolBase& mol,
                  const OESystem::OEUnaryPredicate<OEChem::OEAtomBase>* pred = 0,
                  const double rcut = 0.0) const
bool HasVdwParams(const OEChem::OEMolBase& mol,
                  const OEChem::OEMolBase& host,
                  const double rcut = 0.0) const
```

Returns if the desired Vdw parameters exist for the specified molecule, `mol`. See the *GetVdwParams* method.

HasStretchParams

```
bool HasStretchParams(const OEChem::OEMolBase& mol,
                      const OESystem::OEUnaryPredicate<OEChem::OEAtomBase>* pred = 0)␣
↪const
```

Returns if the desired bond stretching parameters exist for the specified molecule, `mol`. See the *GetStretchParams* method.

HasBendParams

```
bool HasBendParams(const OEChem::OEMolBase& mol,
                   const OESystem::OEUnaryPredicate<OEChem::OEAtomBase>* pred = 0) const
```

Returns if the desired bend parameters exist for the specified molecule, `mol`. See the *GetBendParams* method.

HasTorsionParams

```
bool HasTorsionParams(const OEChem::OEMolBase& mol,
                      const OESystem::OEUnaryPredicate<OEChem::OEAtomBase>* pred = 0)␣
↪const
```

Returns if the desired torison parameters exist for the specified molecule, `mol`. See the *GetTorsionParams* method.

GetTitle

```
std::string GetTitle() const
```

Returns name of the force field parameter.

3.2.5 OEFFPotential

> **Attention**
>
> This API is currently available in C++ and Python.

```
class OEFFPotential : public virtual OEMolPotential::OEMolFunc2
```

The *OEFFPotential* is an abstract base class. This class defines a common base class for all force field potential function implementations.

The following methods are publicly inherited from *OEFunc0*:

- *operator()*
- *GetFComponents*
- *NumVar*

The following methods are publicly inherited from *OEFunc1*:

- *operator()*

The following methods are publicly inherited from *OEFunc2*:

- *operator()*

The following methods are publicly inherited from *OEMolFunc*:

- *Clear*
- *ClearPredicates*
- *GetInteractions*
- *Setup*

The *OEFFPotential* implements the following methods:

- *Set*

The following classes derive from this class:

- *OENonBondPotBase*
- *OEInterNonBondPotBase*
- *OEStretchPotential*
- *OEBendPotential*
- *OETorsionPotential*
- *OEOutOfPlanePotential*

Constructors

```
OEFFPotential(const OEFFParams&)
OEFFPotential(const OEFFPotential&)
```

Constructor and copy constructor.

Assignment operator

```
OEFFPotential& operator=(const OEFFPotential&)
```

Assignment operator.

Set

```
bool Set(const OESystem::OEUnaryPredicate<OEChem::OEAtomBase>&,
         unsigned flag = OEFuncType::All) override
```

This method can be used for interaction-level control of the potential terms. Interactions between two atoms are only included if the OEUnaryPredicate test of all the atoms returns `true`. The *Set* methods must be called before the *Setup* to be effective.

3.2.6 OEForceField

> **Attention**
>
> This API is currently available in C++ and Python.

```
class OEForceField : public OEMolFunc1
```

The *OEForceField* is an abstract base class. This class defines an interface for the collection of force field components. The *OEForceField* extends the *OEMolFunc1* class to allow this being a collection of multiple components. The collection of components into a force field needs to be done concomitant with one or more *OEFFParams* derived classes.

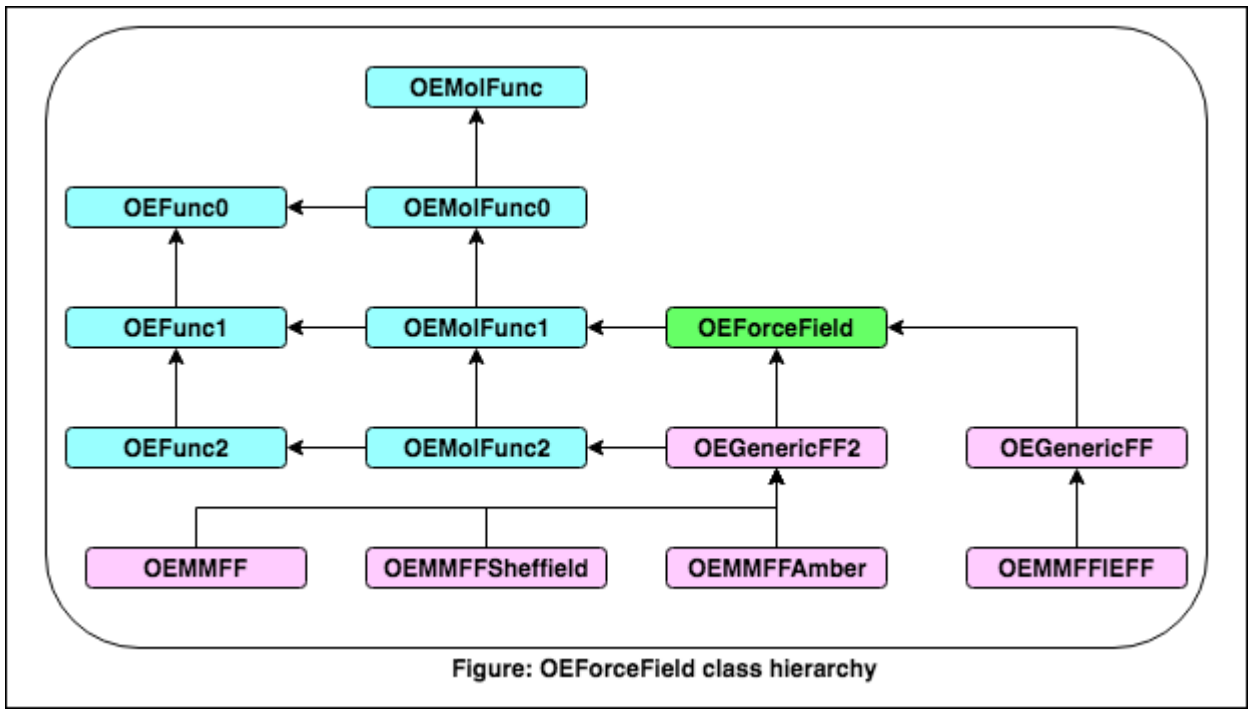


Figure: OEForceField class hierarchy

The following methods are publicly inherited from *OEFunc0*:

- *operator()*
- *GetFComponents*
- *NumVar*

The following methods are publicly inherited from *OEFunc1*:

- *operator()*

The following methods are publicly inherited from *OEMolFunc*:

- *Clear*
- *ClearPredicates*
- *GetInteractions*
- *Setup*

The *OEForceField* class defines the following public methods:

- *GetTitle* and *SetTitle*
- *HasCoulomb*
- *HasInternalCharges*
- *PrepMol*
- *Set*
- *SetNonBondCutoff*

The following classes derive from this class:

- *OEGenericFF*
- *OEGenericFF2*

- *OEMMFF*
- *OEMMFFIEFF*

GetTitle

```
std::string GetTitle() const
```

See *SetTitle*.

HasCoulomb

```
bool HasCoulomb() const
```

Returns if coulombic interactions are currently turned on.

HasInternalCharges

```
bool HasInternalCharges() const
```

Returns True if associated force field parameters has internal charges.

PrepMol

```
bool PrepMol(OEChem::OEMolBase& mol, bool sweep = true,
             bool warnOK = true) const
```

Defines the interface to prepare a molecule to be used for force field setup (*OEMolFunc.Setup*). This method internally calls for all contained *OEFFParams* instances in the *OEForceField* derived class to perform necessary preparation to enable using those parameters.

The second argument to the method specifies whether a call to `OEMolBase.Sweep` is allowed during the preparation. By default, sweeping a molecule is allowed by the method. Methods that override *OEFFParams.PrepMol* may alter the contents of the passed molecule.

Method returns `true` if preparation is successful, and `false` otherwise.

See also

- *Energy components of a single ligand in vacuum* example

Set

```
bool Set(const OESystem::OEUnaryPredicate<OEChem::OEBondBase> &pred,
         unsigned int functype)
bool Set(const OESystem::OEUnaryPredicate<OEChem::OEAtomBase> &pred,
         unsigned int functype)
bool Set(const OESystem::OEBinaryPredicate<OEChem::OEAtomBase,OEChem::OEAtomBase> &pred,
         unsigned int functype)
bool Set(const OESystem::OEBinaryPredicate<OEChem::OEBondBase,OEChem::OEBondBase> &pred,
         unsigned int functype)
```

These methods assigns interaction-control predicates. The first argument is the predicate type to be passed to the interaction component. The second argument is a value or a set of binary OR'ed values taken from the *OEFuncType* namespace which specifies the intended target for the predicate assignment.

The set method passed on the interaction-control predicates to the appropriate force field components contained. Method returns `false` if predicate was not set to any of the components contained by the force field.

Set must be called before the *Setup* for it to be effective.

SetNonBondCutoff

```
bool SetNonBondCutoff(const double)
```

Sets the non-bonded interactions cutoff, in angstroms.

SetTitle

```
void SetTitle(const std::string&)
```

Title for the force field.

3.2.7 OEGenericFF

> **Attention**
>
> This API is currently available in C++ and Python.

```
class OEGenericFF : public OEMolPotential::OEForceField
```

The *OEGenericFF* class facilitates creation of a user defined force field. The force field instance can be extended by adding internally or externally defined potential functions.

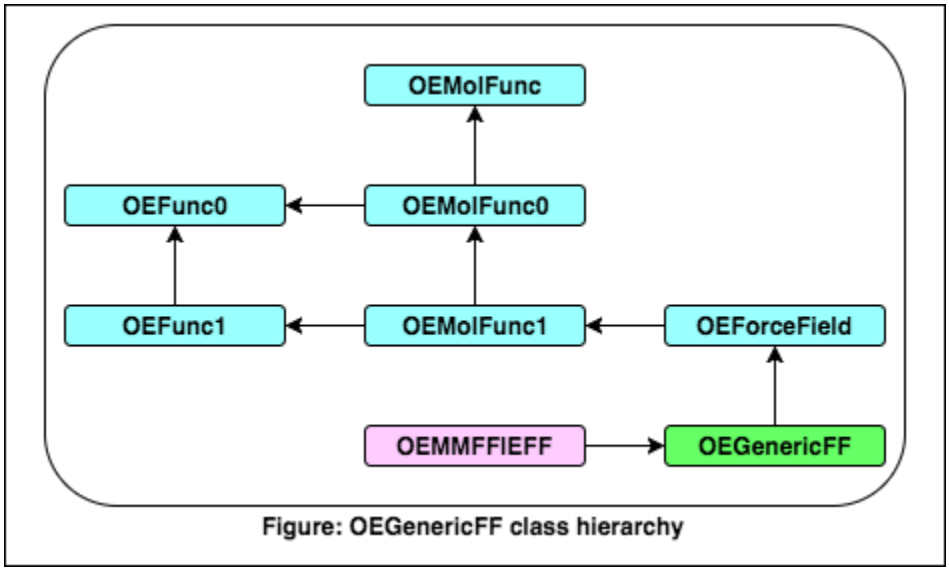


Figure: OEGenericFF class hierarchy

The following methods are publicly inherited from *OEFunc0*:

- *operator()*

- *GetFComponents*
- *NumVar*

The following methods are publicly inherited from *OEFunc1*:

- *operator()*

The following methods are publicly inherited from *OEMolFunc*:

- *Clear*
- *ClearPredicates*
- *GetInteractions*
- *Setup*

The following methods are publicly inherited from *OEForceField*:

- *GetTitle* and *SetTitle*
- *HasCoulomb*
- *HasInternalCharges*
- *PrepMol*
- *Set*
- *SetNonBondCutoff*

The *OEGenericFF* class defines the following public methods:

- *AddMolFunc*
- *GetComponents*
- *RemoveMolFunc*

The following classes derive from this class:

- *OEMMFFIEFF*

Constructors

```
OEGenericFF()
```

Default and copy constructors.

AddMolFunc

```
bool AddMolFunc(OEMolPotential::OEMolFunc1& f, bool own = false)
```

This method defines the interface for adding externally declared functions to an *OEGenericFF* derived instance. The second argument denotes whether the *OEGenericFF* derived instance is to take ownership of the memory occupied by the external function. If ownership is passed, then it would be deleted during destruction of the *OEGenericFF* derived class. Any object derived from the *OEMolFunc1* may be added using this method. The method returns `true` if the object of *OEMolFunc1* is successfully added. Only one copy of a *OEMolFunc1* may be added to an *OEGenericFF* instance. Subsequent attempts to add the same function object to already contained in a *OEGenericFF* derived class will fail with a return value of `false`.

GetComponents

```
std::vector<OEMolPotential::OEMolFunc1*> GetComponents() const
```

Method returns a vector of pointers to the currently contained *OEMolFunc1* instances on this *OEGenericFF*.

RemoveMolFunc

```
bool RemoveMolFunc(OEMolPotential::OEMolFunc1&)
```

Defines an interface to remove a function previously added to an *OEGenericFF* instance. If the function is not contained in the *OEGenericFF* instance then the method will return `false`. If the function is found and removed successfully, then the method will return `true`.

3.2.8 OEGenericFF2

> **Attention**
>
> This API is currently available in C++ and Python.

```
class OEGenericFF2 : public OEMolPotential::OEForceField,  public␣
↪OEMolPotential::OEMolFunc2
```

The *OEGenericFF2* class facilitates creation of a user defined force field of type *OEMolFunc2*. The force field instance can be extended by adding internally or externally defined potential functions.

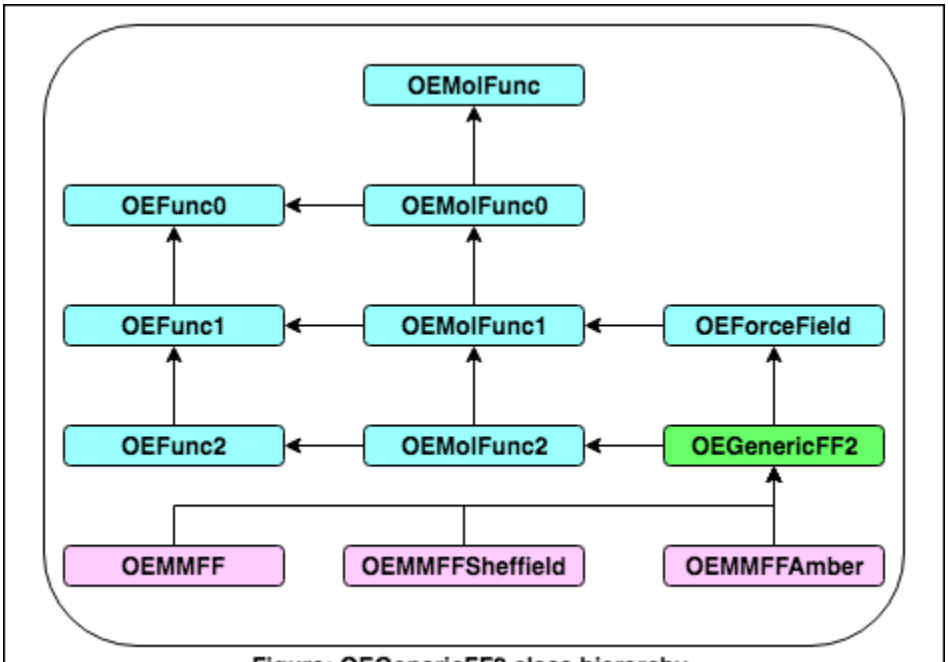


Figure: OEGenericFF2 class hierarchy

The following methods are publicly inherited from *OEFunc0*:

- *operator()*

- *GetFComponents*
- *NumVar*

The following methods are publicly inherited from *OEFunc1*:

- *operator()*

The following methods are publicly inherited from *OEFunc2*:

- *operator()*

The following methods are publicly inherited from *OEMolFunc*:

- *Clear*
- *ClearPredicates*
- *GetInteractions*
- *Setup*

The following methods are publicly inherited from *OEForceField*:

- *GetTitle* and *SetTitle*
- *HasCoulomb*
- *HasInternalCharges*
- *PrepMol*
- *Set*
- *SetNonBondCutoff*

The *OEGenericFF2* class defines the following public methods:

- *AddMolFunc*
- *GetComponents*
- *RemoveMolFunc*

The following classes derive from this class:

- *OEMMFF*

Constructors

```
OEGenericFF2()
```

Default and copy constructors.

AddMolFunc

```
bool AddMolFunc(OEMolPotential::OEMolFunc2& f, bool own = false)
```

This method defines the interface for adding externally declared functions to an *OEGenericFF2* derived instance. The second argument denotes whether the *OEGenericFF2* derived instance is to take ownership of the memory occupied by the external function. If ownership is passed, then it would be deleted during destruction of the *OEGenericFF2* derived class. Any object derived from the *OEMolFunc2* may be added using this method. The method returns `true` if the object of *OEMolFunc2* is successfully added. Only one copy of a *OEMolFunc2* may be added to an *OEGenericFF2* instance. Subsequent attempts to add the same function object to already contained in a *OEGenericFF2* derived class will fail with a return value of `false`.

GetComponents

```
std::vector<OEMolPotential::OEMolFunc2*> GetComponents() const
```

Method returns a vector of pointers to the currently contained *OEMolFunc2* instances on this *OEGenericFF2*.

RemoveMolFunc

```
bool RemoveMolFunc(OEMolPotential::OEMolFunc2&)
```

Defines an interface to remove a function previously added to an *OEGenericFF2* instance. If the function is not contained in the *OEGenericFF2* instance then the method will return `false`. If the function is found and removed successfully, then the method will return `true`.

3.2.9 OEInteraction

> **Attention**
>
> This API is currently available in C++ and Python.

```
class OEInteraction
```

The *OEInteraction* class provides a general mechanism to report individual interactions calculated within a function or force field component. For instance, pairwise atom interactions in an intramolecular VdW component calculation can be obtained by the *OEMolFunc.GetInteractions* method that returns an iterator over the *OEInteraction* instances.

The *OEInteraction* class defines the following public methods:

- *GetAtoms*
- *GetName*
- *GetValues*
- *NumAtoms*
- *NumValues*

Constructors

```
OEInteraction(OEInteractImpl &)
OEInteraction(const OEInteraction &)
```

Default and copy constructors.

GetAtoms

```
OESystem::OEIterBase<OEChem::OEAtomBase>* GetAtoms() const
```

Returns an iterator over the atoms involved in the calculated interaction.

GetName

```
const char* GetName() const
```

Returns a clear text human readable identifier for the type of interaction.

GetValues

```
double GetValues(double* grads = 0) const
```

Returns the energy associated with the calculated interaction. The associated gradients may also be obtained by passing in an array of the length returned by the *OEInteraction.NumValues* method. The derivatives correspond to the atom ordering returned by the *OEInteraction.GetAtoms* method.

NumAtoms

```
unsigned NumAtoms() const
```

Returns the number of atoms involved in the calculated interactions. One-body interactions, such as an atom in a field, will only have one atom involved in the interaction. VdW interactions will consist of two atoms. Torsions are composed of four interacting atoms.

NumValues

```
unsigned NumValues() const
```

Returns the number of gradients associated with the particular interaction. For interactions calculated from Cartesian coordinates of atoms, this method will typically return three times the number of atoms involved in the interaction as x, y, and z gradient values are calculated for each atom.

3.2.10 OEInteractParams

> **Attention**
>
> This API is currently available in C++ and Python.

```
class OEInteractParams
```

The *OEInteractParams* class provides a general mechanism for retrieving multiple parameters from methods defined in the *OEFFParams* class. The number and types of parameters required for calculating molecular interactions varies among force fields. To keep the interface consistent among implementations of *OEFFParams* for various force fields, parameters are stored in the *OEInteractParams* class. Once parameters have been assigned in an *OEInteractParams* instance they may be retrieved using *OEInteractParams.Get* method.

The *OEInteractParams* class defines the following public methods:

- *Clear*
- *Get*
- *GetList*
- *Set*
- *SetList*

Constructors

```
OEInteractParams()
OEInteractParams(const OEInteractParams&)
```

Constructor and copy constructor.

Assignment operator

```
OEInteractParams& operator=(const OEInteractParams&)
```

Thwe assignment operator

Clear

```
void Clear()
```

Assigns all parameters to 0.0 in the *OEInteractParams* instance.

Get

```
double Get(const unsigned int idx) const
```

Returns the parameter value associated with an integer taken from *OEFFParam* namespace. The integer argument specifies the type of parameter requested from the *OEInteractParams* instance. The requested parameter type must match *OEFFParams* retrieval method to which the *OEInteractParams* instance was passed.

GetList

```
std::vector<double> GetList(const unsigned idx) const
```

Returns a vector of parameters associated with an integer taken from *OEFFParam* namespace. The integer argument specifies the type of parameter vector requested from the *OEInteractParams* instance.

Set

```
bool Set(const unsigned idx, const double value)
```

Sets the value of a parameter associated with an integer taken from the *OEFFParam* namespace (first argument) in the *OEFFParam* namespace instance.

SetList

```
bool SetList(const unsigned idx, const std::vector<double>& values)
```

Sets the values of a parameter vector in the *OEFFParam* namespace instance. The integer argument specifies the type of parameter vector which will be set.

3.2.11 OEInterAdaptor

Attention

This API is currently available in C++ and Python.

```
class OEInterAdaptor : public OEMolAdaptor
```

The *OEInterAdaptor* class provides the ability to optimize a molecule or conformation with the use of internal coordinates (bond lengths, bond angle bonds and dihedral angles) defined by the internal Z-matrix.

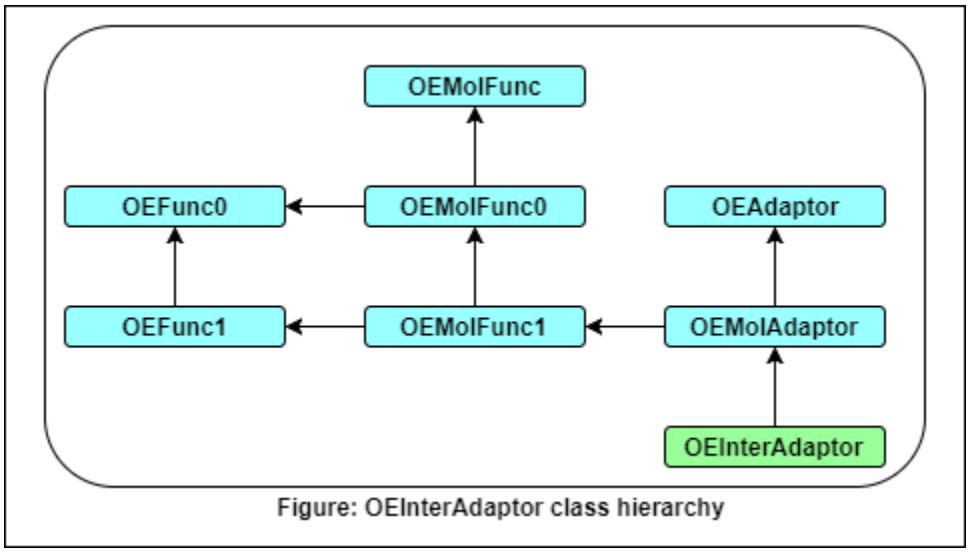


Figure: OEInterAdaptor class hierarchy

The following methods are publicly inherited from *OEAdaptor*:

- *AdaptVar*
- *GetVar*

The following methods are publicly inherited from *OEFunc0*:

- *operator()*
- *GetFComponents*
- *NumVar*

The following methods are publicly inherited from *OEFunc1*:

- *operator()*

The following methods are publicly inherited from *OEMolFunc*:

- *Clear*
- *ClearPredicates*
- *GetInteractions*
- *Setup*

The following methods are publicly inherited from *OEMolAdaptor*:

- *SetMolFunc*

The *OEInterAdaptor* class defines the following public methods:

- *AdaptGrad*
- *Set*

Constructors

```
OEInterAdaptor(OEMolFunc1 &, bool dihedrals = true,
               bool own = false)
```

The molecule function used for function evaluation must be provided as the first argument to the constructor. The second argument allows to fix all dihedral angles. By default all dihedral angles are optimized. The third argument specifies whether the *OEInterAdaptor* object takes ownership of the memory of the molecule function instance. By default the *OEInterAdaptor* instance does not take ownership of the molecule function, so the *OEInterAdaptor* destructor does not delete the molecule function instance. If ownership of the molecule function is transferred to the *OEInterAdaptor* instance, the molecule function's delete operator will be called in the *OEInterAdaptor* destructor.

Default and copy constructors.

AdaptGrad

```
bool AdaptGrad(double* ga, const double* g) const
```

Transforms the set of Cartesian gradients passed as a second argument to the method, into a set of internal coordinates gradients and copies them into the array passed as a first argument.

Set

```
bool Set(const OESystem::OEUnaryPredicate<OEChem::OEAtomBase>&,
         unsigned int functype = OEFuncType::InterAdapt)
bool Set(const OESystem::OEUnaryPredicate<OEChem::OEBondBase>&,
         unsigned int functype = OEFuncType::InterAdapt)
```

Those two methods assign atom and bond predicates which are used to partially restrict molecules during optimization. First method specifies atoms which are to be held fixed in their starting positions. Second method is used to fix dihedral angles a1-a2-a3-a4 defined by a middle bond a2-a3. Both methods *Set* must be called before the method derived from *Setup*.

3.2.12 OEInterCoulombPotential

> **Attention**
>
> This API is currently available in C++ and Python.

```
class OEInterCoulombPotential : public OEInterNonBondBase
```

The *OEInterCoulombPotential* class defines an interface for calculation of inter-molecular Coulomb interaction potentials.

The following methods are publicly inherited from *OEFunc0*:

- *operator()*
- *GetFComponents*
- *NumVar*

The following methods are publicly inherited from *OEFunc1*:

- *operator()*

The following methods are publicly inherited from *OEFunc2*:

- *operator()*

The following methods are publicly inherited from *OEMolFunc*:

- *Clear*
- *ClearPredicates*
- *GetInteractions*
- *Setup*

The following methods are publicly inherited from *OEFFPotential*:

- *Set*

The following methods are publicly inherited from *OEInterMolFunc1*:

- *AdaptHostCoords*
- *GetVar*
- *SetHostFlex*
- *SetupHost*

The following methods are publicly inherited from *OEInterNonBondPotBase*:

- *Set*
- *SetNonBondCutoff*

Constructors

```
OEInterCoulombPotential(const OEMolPotential::OEFFParams& fp,
                        const double rcut = 0.0)
OEInterCoulombPotential(const OEInterCoulombPotential&)
```

Constructor and copy constructor.

Assignment operator

```
OEInterCoulombPotential& operator=(const OEInterCoulombPotential&)
```

Assignment operator.

3.2.13 OEInterMolFunc1

> **Attention**
>
> This API is currently available in C++ and Python.

```
class OEInterMolFunc1 : public OEMolFunc1
```

The *OEInterMolFunc1* is an abstract base class. It extends *OEMolFunc1* to the cases where a small molecule interacts with a macromolecule that is rigid or partially flexible.

The following methods are publicly inherited from *OEFunc0*:

- *operator()*
- *GetFComponents*
- *NumVar*

The following methods are publicly inherited from *OEFunc1*:

- *operator()*

The following methods are publicly inherited from *OEMolFunc*:

- *Clear*
- *ClearPredicates*
- *GetInteractions*
- *Set*
- *Setup*

The *OEInterMolFunc1* class defines the following public methods:

- *AdaptHostCoords*
- *GetVar*
- *SetHostFlex*
- *SetupHost*

The following classes derive from this class:

- *OEInterMolFunc2*

AdaptHostCoords

```
bool AdaptHostCoords(double* coords, const double* var) const
```

Produces the complete host coordinates (`coords`) from a set of host coordinates representing the flexible part of the host (`var`).

GetVar

```
void GetVar(double* var, const double* x)
```

Takes the complete set coordinates of the host-ligand system (`x`) and transforms them to a subset of coordinates to be optimized (`var`).

SetHostFlex

```
bool SetHostFlex(const OESystem::OEUnaryPredicate<OEChem::OEAtomBase>& pred)
```

Sets predicate (OEUnaryPredicate) that defines the flexible atoms of the host molecule (usually a macromolecule). Host atoms that pass the predicate (OEUnaryPredicate) will be subject to optimization along with the ligand atoms for which a *Setup* method is called.

SetupHost

```
bool SetupHost(const OEChem::OEMolBase& host)
```

Sets a macromolecule as host.

3.2.14 OEInterMolFunc2

> **Attention**
>
> This API is currently available in C++ and Python.

```
class OEInterMolFunc2 : public InterMolFunc1, public OEMolFunc2
```

The *OEInterMolFunc2* is an abstract base class. This class defines the interface for stateful functions which operate on molecules and conformers, compute a function value, gradients, and second derivatives. *OEInterMolFunc2* combines *OEInterMolFunc1* with *OEMolFunc2* interface. Classes derived from *OEInterMolFunc2* can be used with any optimizers.

The following methods are publicly inherited from *OEFunc0*:

- *operator()*
- *GetFComponents*
- *NumVar*

The following methods are publicly inherited from *OEFunc1*:

- *operator()*

The following methods are publicly inherited from *OEFunc2*:

- *operator()*

The following methods are publicly inherited from *OEMolFunc*:

- *Clear*
- *ClearPredicates*
- *GetInteractions*
- *Set*
- *Setup*

The following methods are publicly inherited from *OEInterMolFunc1*:

- *AdaptHostCoords*
- *GetVar*
- *SetHostFlex*
- *SetupHost*

The following classes derive from this class:

- *OEInterNonBondPotBase*

3.2.15 OEInterNonBondPotBase

> **Attention**
>
> This API is currently available in C++ and Python.

```
class OEInterNonBondPotBase : public OEFFPotential, public␣
↪OEMolPotential::OEInterMolFunc2
```

The *OEInterNonBondPotBase* is an abstract base class. This class defines a common base class for all inter-molecular non-bonded force field interaction potentials.

The following methods are publicly inherited from *OEFunc0*:

- *operator()*
- *GetFComponents*
- *NumVar*

The following methods are publicly inherited from *OEFunc1*:

- *operator()*

The following methods are publicly inherited from *OEFunc2*:

- *operator()*

The following methods are publicly inherited from *OEMolFunc*:

- *Clear*
- *ClearPredicates*
- *GetInteractions*
- *Setup*

The following methods are publicly inherited from *OEFFPotential*:

- *Set*

The following methods are publicly inherited from *OEInterMolFunc1*:

- *AdaptHostCoords*
- *GetVar*
- *SetHostFlex*
- *SetupHost*

The *OEInterNonBondPotBase* implements the following methods:

- *Set*
- *SetNonBondCutoff*

The following classes derive from this class:

- *OEInterNonBondPotential*
- *OEInterVdwPotential*
- *OEInterCoulombPotential*

Constructors

```
OEInterNonBondPotBase(const OEMolPotential::OEFFParams& fp,
                      const OENonBondIntcsOptions& options = OENonBondIntcsOptions())
OEInterNonBondPotBase(const OEInterNonBondPotBase&)
```

Constructor and copy constructor.

Assignment operator

```
OEInterNonBondPotBase& operator=(const OEInterNonBondPotBase&)
```

Assignment operator.

Set

```
bool Set(const OESystem::OEBinaryPredicate<OEChem::OEAtomBase, OEChem::OEAtomBase>& pred,
         unsigned int flag = OEFuncType::All)
```

This method can be used for interaction-level control of the non-bonded terms. Interactions between two atoms are only included if the test of the binary predicate with both atoms returns `true`. The *Set* method must be called before the *Setup* for it to be effective.

SetNonBondCutoff

```
bool SetNonBondCutoff(const double)
```

Sets non-bonded interactions cutoff. Setting a value of `0.0` or less assumes that all interactions should be included without any cutoff.

3.2.16 OEInterNonBondPotential

> **Attention**
>
> This API is currently available in C++ and Python.

```
class OEInterNonBondPotential : public OEInterNonBondBase
```

The *OEInterNonBondPotential* class defines an interface for calculation of inter-molecular non-bonded (both VdW and Coulomb) force field interaction potentials..

The following methods are publicly inherited from *OEFunc0*:

- *operator()*
- *GetFComponents*
- *NumVar*

The following methods are publicly inherited from *OEFunc1*:

- *operator()*

The following methods are publicly inherited from *OEFunc2*:

- *operator()*

The following methods are publicly inherited from *OEMolFunc*:

- *Clear*
- *ClearPredicates*
- *GetInteractions*
- *Setup*

The following methods are publicly inherited from *OEFFPotential*:

- *Set*

The following methods are publicly inherited from *OEInterMolFunc1*:

- *AdaptHostCoords*
- *GetVar*
- *SetHostFlex*
- *SetupHost*

The following methods are publicly inherited from *OEInterNonBondPotBase*:

- *Set*
- *SetNonBondCutoff*

Constructors

```
OEInterNonBondPotential(const OEMolPotential::OEFFParams& fp,
                        const OENonBondIntcsOptions& options = OENonBondIntcsOptions())
OEInterNonBondPotential(const OEInterNonBondPotential&)
```

Constructor and copy constructor.

Assignment operator

```
OEInterNonBondPotential& operator=(const OEInterNonBondPotential&)
```

Assignment operator.

3.2.17 OEInterVdwPotential

> **Attention**
>
> This API is currently available in C++ and Python.

```
class OEInterVdwPotential : public OEInterNonBondBase
```

The *OEInterVdwPotential* class defines an interface for calculation of inter-molecular Vdw force field interaction potentials.

The following methods are publicly inherited from *OEFunc0*:

- *operator()*
- *GetFComponents*

- *NumVar*

The following methods are publicly inherited from *OEFunc1*:

- *operator()*

The following methods are publicly inherited from *OEFunc2*:

- *operator()*

The following methods are publicly inherited from *OEMolFunc*:

- *Clear*
- *ClearPredicates*
- *GetInteractions*
- *Setup*

The following methods are publicly inherited from *OEFFPotential*:

- *Set*

The following methods are publicly inherited from *OEInterMolFunc1*:

- *AdaptHostCoords*
- *GetVar*
- *SetHostFlex*
- *SetupHost*

The following methods are publicly inherited from *OEInterNonBondPotBase*:

- *Set*
- *SetNonBondCutoff*

Constructors

```
OEInterVdwPotential(const OEMolPotential::OEFFParams& fp,
                    const double rcut = 0.0)
OEInterVdwPotential(const OEInterVdwPotential&)
```

Constructor and copy constructor.

Assignment operator

```
OEInterVdwPotential& operator=(const OEInterVdwPotential&)
```

Assignment operator.

3.2.18 OEMolAdaptor

> **Attention**
>
> This API is currently available in C++ and Python.

```
class OEMolAdaptor : public OEOpt::OEAdaptor, public OEMolPotential:: OEMolFunc1
```

The *OEMolAdaptor* class is a pure virtual base class which defines the interface for classes which adapt or transform molecular coordinates and gradients during an optimization. Adaptors are useful in cases where energy and gradients evaluation for a molecule or conformation are performed in Cartesian coordinates, but the optimization is done in different coordinate system, for example in torsion space.

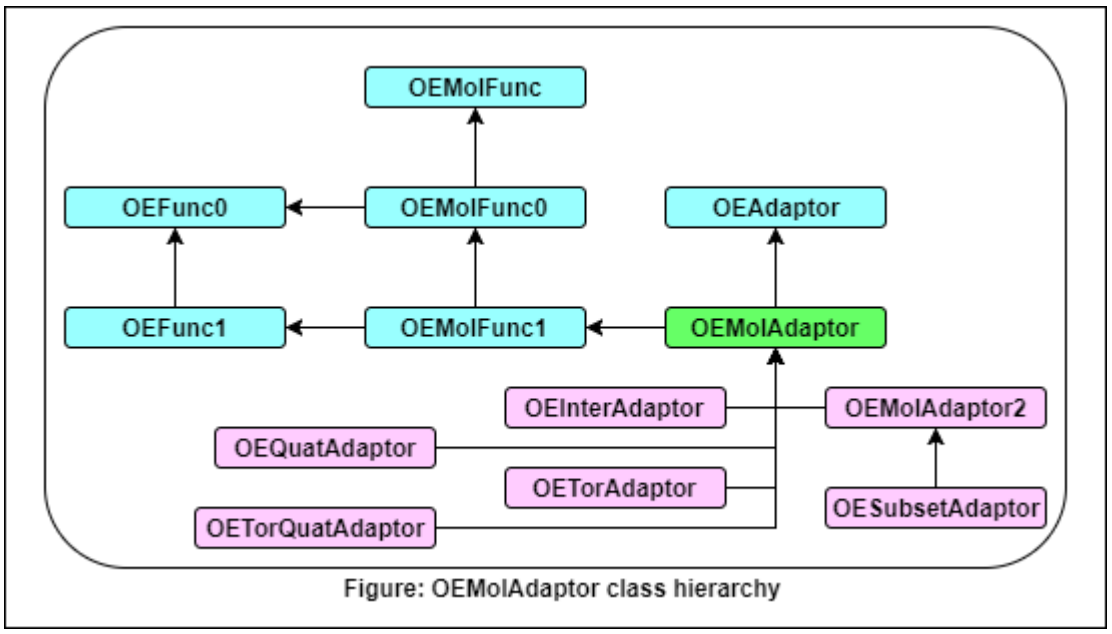


Figure: OEMolAdaptor class hierarchy

The following methods are publicly inherited from *OEAdaptor*:

- *AdaptVar*
- *GetVar*

The following methods are publicly inherited from *OEFunc0*:

- *operator()*
- *GetFComponents*
- *NumVar*

The following methods are publicly inherited from *OEFunc1*:

- *operator()*

The following methods are publicly inherited from *OEMolFunc*:

- *Clear*
- *ClearPredicates*
- *GetInteractions*
- *Set*
- *Setup*

The *OEMolAdaptor* class defines the following public methods:

- *SetMolFunc*

The following classes derive from this class:

- *OEInterAdaptor*

- *OEMolAdaptor2*
- *OEQuatAdaptor*
- *OESubsetAdaptor*
- *OETorAdaptor*
- *OETorQuatAdaptor*

SetMolFunc

```
bool SetMolFunc(OEMolFunc1& molfunc, bool own)=0
```

Defines the interface for resetting the molecule function within a *OEMolAdaptor* derived instance. Any molecule functions which are "owned" at the time of calling this method will be deleted. If the value of a second argument passed to the method is `true`, the adaptor derived from *OEMolAdaptor* will take ownership of the memory of the molecule function, and will call the molecule function's delete operator when either the destructor or the current method is called.

3.2.19 OEMolAdaptor2

> **Attention**
>
> This API is currently available in C++ and Python.

```
class OEMolAdaptor2 : public OEMolPotential:: OEMolAdaptor, public OEMolPotential::␣
↪OEMolFunc2
```

The *OEMolAdaptor2* class is a pure virtual base class which defines the interface for classes which adapt or transform molecular coordinates, gradients, and hessians of a given *OEMolFunc2* during an optimization. Adaptors are useful in cases where energy, gradients and hessians evaluation for a molecule or conformation are performed in Cartesian coordinates, but the optimization is done in different coordinate system.

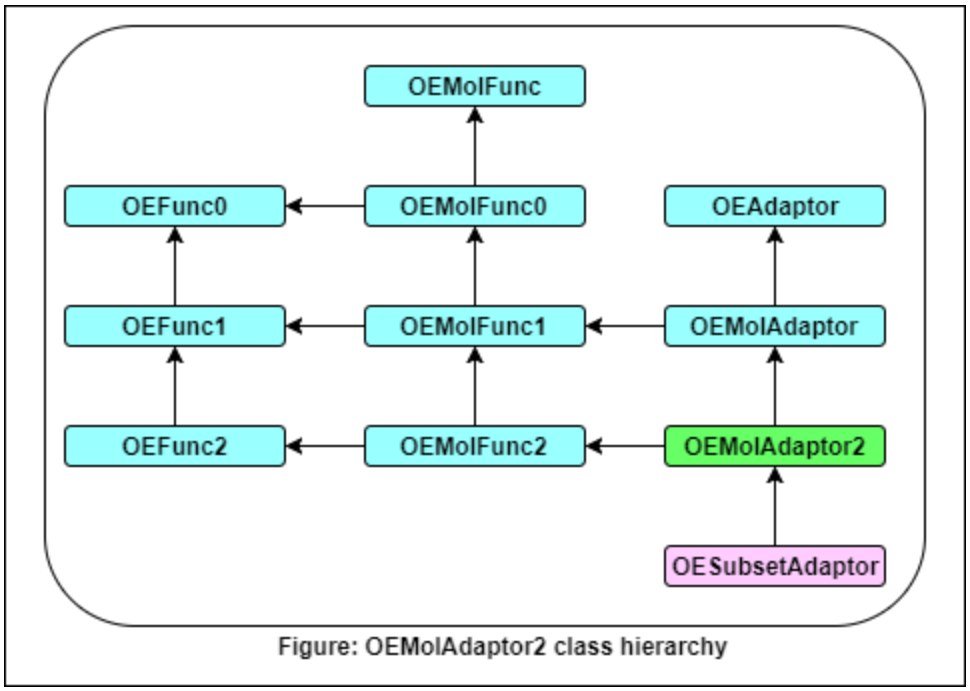


Figure: OEMolAdaptor2 class hierarchy

The following methods are publicly inherited from *OEAdaptor*:

- *AdaptVar*
- *GetVar*

The following methods are publicly inherited from *OEFunc0*:

- *operator()*
- *GetFComponents*
- *NumVar*

The following methods are publicly inherited from *OEFunc1*:

- *operator()*

The following methods are publicly inherited from *OEFunc2*:

- *operator()*

The following methods are publicly inherited from *OEMolFunc*:

- *Clear*
- *ClearPredicates*
- *GetInteractions*
- *Set*
- *Setup*

The following methods are publicly inherited from *OEMolAdaptor*:

- *SetMolFunc*

The *OEMolAdaptor2* class defines the following public methods:

- *SetMolFunc2*

The following classes derive from this class:

- *OESubsetAdaptor*

SetMolFunc2

```
bool SetMolFunc2(OEMolFunc2& molfunc, bool own)=0
```

Defines the interface for resetting the molecule function, *OEMolFunc2*, within a *OEMolAdaptor2* derived instance. Any molecule functions which are "owned" at the time of calling this method will be deleted. If the value of a second argument passed to the method is `true`, the adaptor derived from *OEMolAdaptor2* will take ownership of the memory of the molecule function, and will call the molecule function's delete operator when either the destructor or the current method is called.

3.2.20 OEMolFunc

> **Attention**
>
> This API is currently available in C++ and Python.

class OEMolFunc

The *OEMolFunc* is an abstract base class. This class defines the interface for functions that require a molecule to compute the function value, and is often associated with a force field. When associated with a force field, this represents a forcefield component or a collection of components with each component being a collection of interactions (*OEInteraction*). Thus, classes derived from *OEMolFunc* can represent a part of or a complete force field. The force field components need to be done concomitant with one or more *OEFFParams* derived classes.

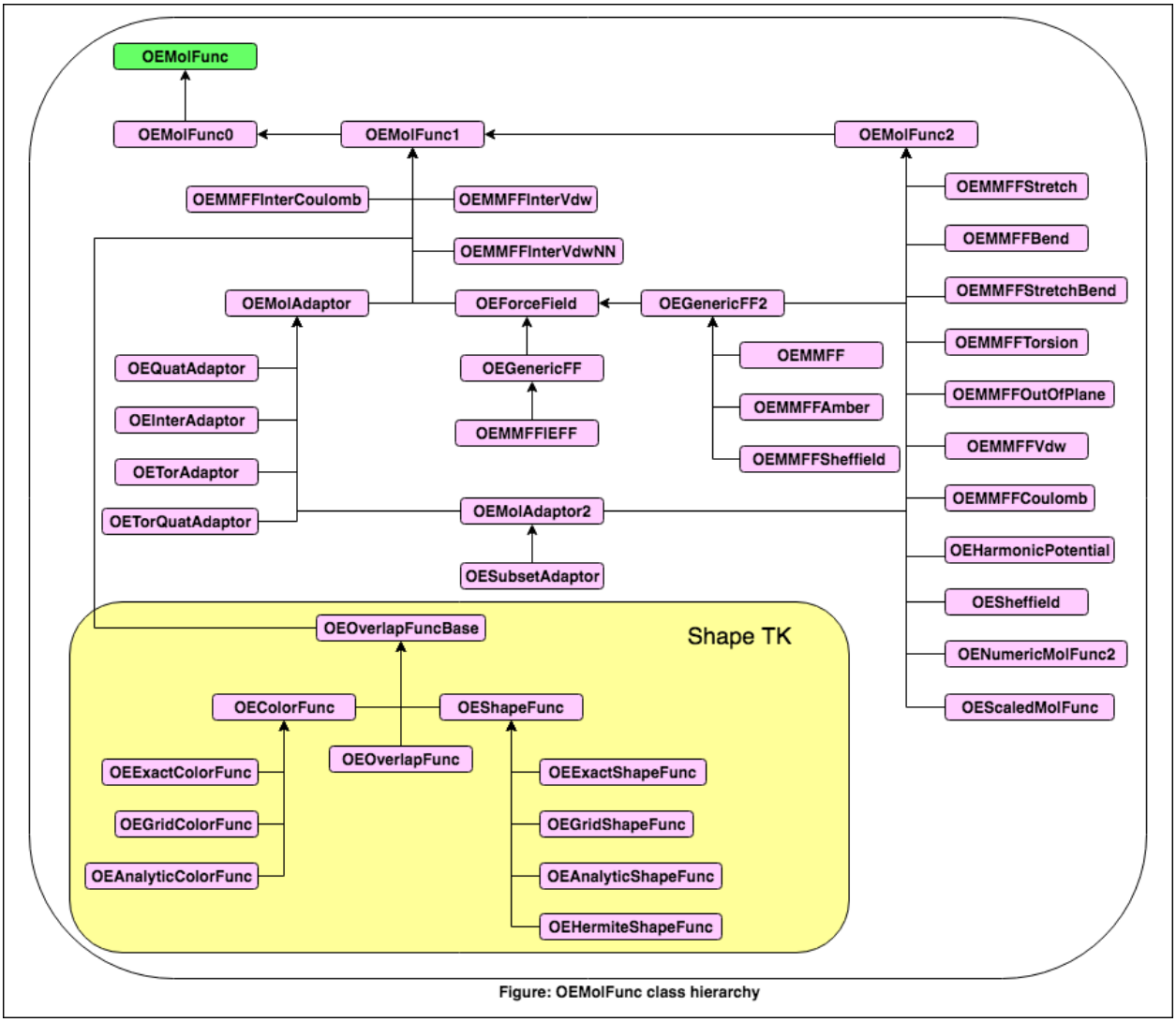


Figure: OEMolFunc class hierarchy

The *OEMolFunc* class defines the following public methods:

- *Clear*
- *ClearPredicates*
- *GetInteractions*

- *Set*
- *Setup*
- *SetupConf*

The following classes derive from this class:

- *OEMolFunc0*
- *OEMolFunc1*
- *OEMolFunc2*
- *OEForceField*
- *OEGenericFF*
- *OEGenericFF2*
- *OEInterAdaptor*
- *OEMolAdaptor*
- *OEQuatAdaptor*
- *OESubsetAdaptor*
- *OETorAdaptor*
- *OETorQuatAdaptor*
- *OENumericMolFunc2*
- *OEScaledMolFunc*
- *OEFFPotential*
- *OEStretchPotential*
- *OEBendPotential*
- *OETorsionPotential*
- *OEOutOfPlanePotential*
- *OENonBondPotBase*
- *OEVdwPotential*
- *OECoulombPotential*
- *OEInterMolFunc1*
- *OEInterMolFunc2*
- *OEInterNonBondPotBase*
- *OEInterNonBondPotential*
- *OEInterVdwPotential*
- *OEInterCoulombPotential*
- *OEHarmonicPotential*
- *OEMMFF*
- *OEMMFFStretchBend*
- *OEMMFFIEFF*

- *OESheffield*
- OEOverlapFuncBase
- OEOverlapFunc
- OEShapeFunc
- OEExactShapeFunc
- OEAnalyticShapeFunc
- OEHermiteShapeFunc
- OEGridShapeFunc
- OEColorFunc
- OEExactColorFunc
- OEAnalyticColorFunc
- OEGridColorFunc

Clear

```
void Clear()
```

Deletes all interactions contained in the *OEMolFunc* derived instance. The interactions are created during setup (*OEMolFunc.Setup*).

ClearPredicates

```
void ClearPredicates()
```

Deletes all stored predicates set, previously created by using one of the overloaded *OEMolFunc.Set* methods. Predicates control function behavior during setup (*OEMolFunc.Setup*).

GetInteractions

```
OESystem::OEIterBase<OEInteraction>* GetInteractions(const OEChem::OEMolBase &,
                                                     const double *,
                                                     unsigned functype = 0) const
```

This method defines the interface for retrieving interaction level information computed by the *OEMolFunc* instance, given a molecule (first argument) and a set of coordinates (second argument). The method returns an iterator over the *OEInteraction* objects. See the section on *OEInteraction* for the details regarding interaction interrogation and reporting.

If *OEMolFunc* is a composition of multiple components, the last argument may be used to request the interactions of only a single component by passing a value from the *OEFuncType* namespace.

Due to the unique construct of the return value from this method, it is not possible to implement this in a user derived class. Default implementation of this method return an iterator over an empty vector.

Set

```
bool Set(const OESystem::OEUnaryPredicate<OEChem::OEBondBase>& pred,
         unsigned int functype)
bool Set(const OESystem::OEUnaryPredicate<OEChem::OEAtomBase>& pred,
```

(continues on next page)

(continued from previous page)

```
        unsigned int functype)
bool Set(const OESystem::OEBinaryPredicate<OEChem::OEAtomBase, OEChem::OEAtomBase>& pred,
        unsigned int functype)
bool Set(const OESystem::OEBinaryPredicate<OEChem::OEBondBase, OEChem::OEBondBase>& pred,
        unsigned int functype)
```

These methods assigns interaction-control predicates to the *OEMolFunc* derived instance. The first argument is the predicate type to be passed to the interaction component. The second argument is a value or a set of binary OR'ed values taken from the *OEFuncType* namespace which specifies the intended target for the predicate assignment.

Most derived classes can only use a handful of the four different kinds of predicates allowed in these overloads, thus only the specific overloads are overwritten by various derived class. The default implementation for all the overloads has trivial implementation and returns a `false`.

Setup

```
bool Setup(const OEChem::OEMolBase&)
```

This method defines the interface for initializing a *OEMolFunc* derived instance with a molecule. All classes derived from *OEMolFunc* implements this method. Typically the implementation of this method requires perception of the passed molecule for efficient energy evaluation.

See also

- *Energy components of a single ligand in vacuum* example

SetupConf

```
bool SetupConf(const double *)
```

This method defines the interface for initializing a *OEMolFunc* derived instance with coordinates of a new conformer, after it has already been initialized with the molecule by calling *Setup*. Not all classes derived from *OEMolFunc* implements this method, but calling this should never cause any additional performance issue or failures.

3.2.21 OEMolFunc0

Attention

This API is currently available in C++ and Python.

```
class OEMolFunc0 : public OEOpt::OEFunc0, public OEMolFunc
```

The *OEMolFunc0* is an abstract base class. This class defines the interface for stateful functions which operate on molecules and conformers, compute a function value, but have no defined analytical gradients. *OEMolFunc0* combines *OEFunc0* with *OEMolFunc* interface. Classes derived from *OEMolFunc0* are intended for use with optimizers which do not require gradients.

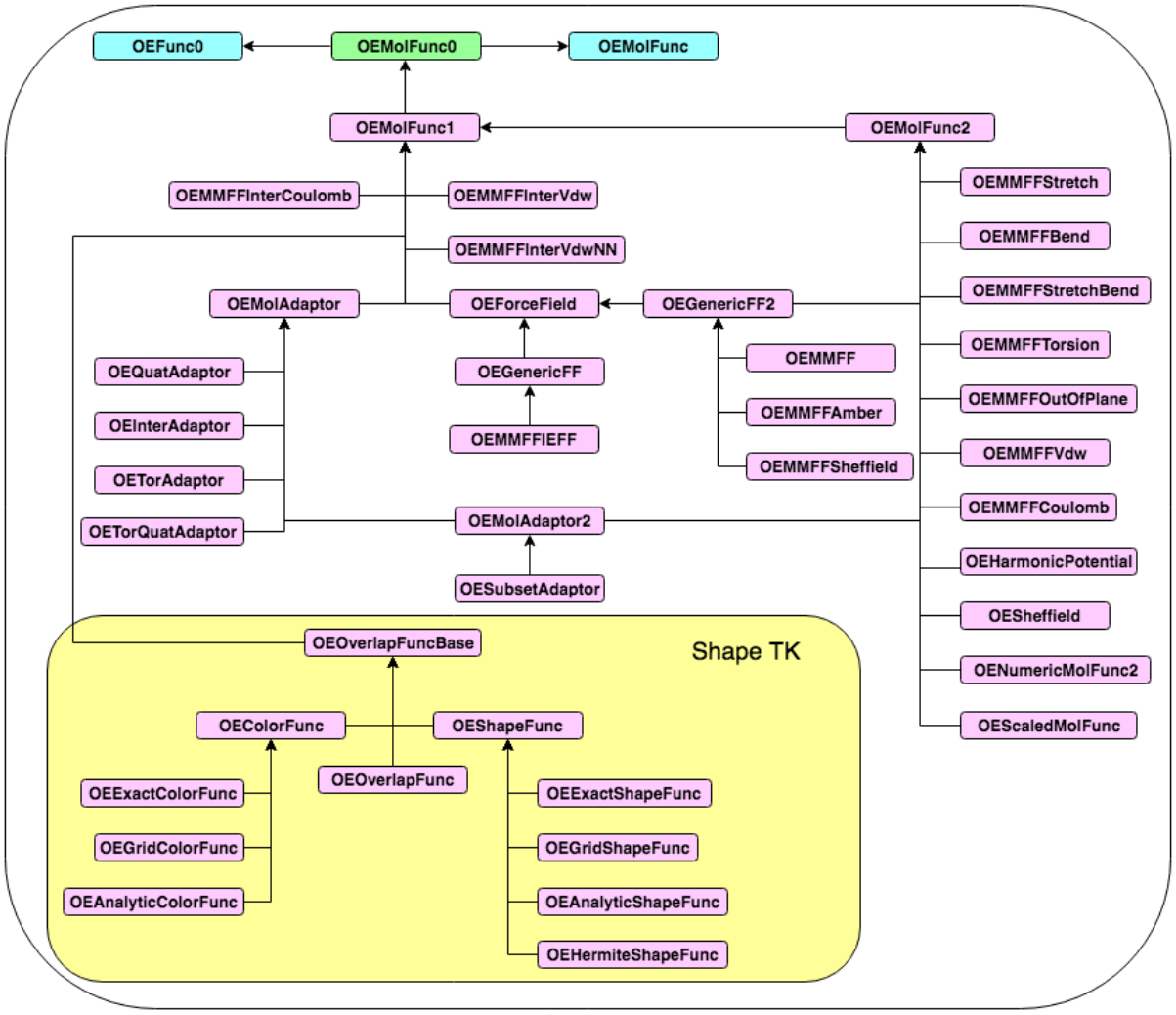


Figure: OEFMolFunc0 class hierarchy

The following methods are publicly inherited from *OEFunc0*:

- *operator()*
- *GetFComponents*
- *NumVar*

The following methods are publicly inherited from *OEMolFunc*:

- *Clear*
- *ClearPredicates*
- *GetInteractions*
- *Set*
- *Setup*

The following classes derive from this class:

- *OEMolFunc1*
- *OEMolFunc2*

- *OEForceField*
- *OEGenericFF*
- *OEGenericFF2*
- *OEInterAdaptor*
- *OEMolAdaptor*
- *OEQuatAdaptor*
- *OESubsetAdaptor*
- *OETorAdaptor*
- *OETorQuatAdaptor*
- *OENumericMolFunc2*
- *OEScaledMolFunc*
- *OEFFPotential*
- *OEStretchPotential*
- *OEBendPotential*
- *OETorsionPotential*
- *OEOutOfPlanePotential*
- *OENonBondPotBase*
- *OEVdwPotential*
- *OECoulombPotential*
- *OEInterMolFunc1*
- *OEInterMolFunc2*
- *OEInterNonBondPotBase*
- *OEInterNonBondPotential*
- *OEInterVdwPotential*
- *OEInterCoulombPotential*
- *OEHarmonicPotential*
- *OEMMFF*
- *OEMMFFStretchBend*
- *OEMMFFIEFF*
- *OESheffield*
- OEOverlapFuncBase
- OEOverlapFunc
- OEShapeFunc
- OEExactShapeFunc
- OEAnalyticShapeFunc
- OEHermiteShapeFunc

- OEGridShapeFunc
- OEColorFunc
- OEExactColorFunc
- OEAnalyticColorFunc
- OEGridColorFunc

3.2.22 OEMolFunc1

> **Attention**
>
> This API is currently available in C++ and Python.

```
class OEMolFunc1 : public OEOpt::OEFunc1, public OEMolFunc0
```

The *OEMolFunc1* is an abstract base class. This class defines the interface for stateful functions which operate on molecules and conformers, compute a function value, and has defined analytical gradients. *OEMolFunc1* combines *OEFunc1* with *OEMolFunc0* interface. Classes derived from *OEMolFunc1* are intended for use with optimizers which require gradients but do not require second derivatives.

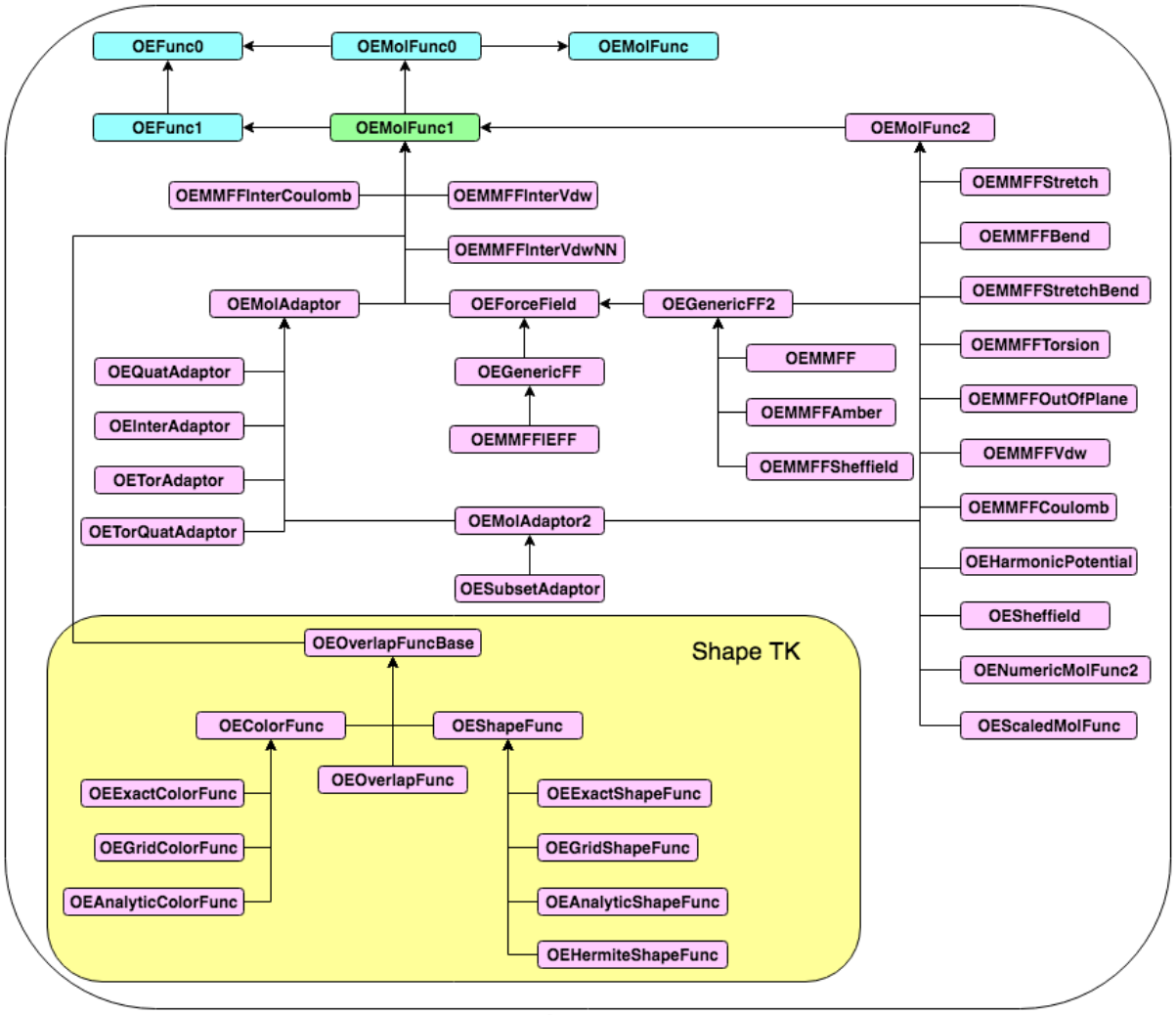


Figure: OEFMolFunc1 class hierarchy

The following methods are publicly inherited from *OEFunc0*:

- *operator()*
- *GetFComponents*
- *NumVar*

The following methods are publicly inherited from *OEFunc1*:

- *operator()*

The following methods are publicly inherited from *OEMolFunc*:

- *Clear*
- *ClearPredicates*
- *GetInteractions*
- *Set*
- *Setup*

The following classes derive from this class:

- *OEMolFunc2*
- *OEForceField*
- *OEGenericFF*
- *OEGenericFF2*
- *OEInterAdaptor*
- *OEMolAdaptor*
- *OEQuatAdaptor*
- *OESubsetAdaptor*
- *OETorAdaptor*
- *OETorQuatAdaptor*
- *OENumericMolFunc2*
- *OEScaledMolFunc*
- *OEFFPotential*
- *OEStretchPotential*
- *OEBendPotential*
- *OETorsionPotential*
- *OEOutOfPlanePotential*
- *OENonBondPotBase*
- *OEVdwPotential*
- *OECoulombPotential*
- *OEInterMolFunc1*
- *OEInterMolFunc2*
- *OEInterNonBondPotBase*
- *OEInterNonBondPotential*
- *OEInterVdwPotential*
- *OEInterCoulombPotential*
- *OEHarmonicPotential*
- *OEMMFF*
- *OEMMFFStretchBend*
- *OEMMFFIEFF*
- *OESheffield*
- OEOverlapFuncBase
- OEOverlapFunc
- OEShapeFunc
- OEExactShapeFunc
- OEAnalyticShapeFunc

- OEHermiteShapeFunc
- OEGridShapeFunc
- OEColorFunc
- OEExactColorFunc
- OEAnalyticColorFunc
- OEGridColorFunc

3.2.23 OEMolFunc2

> **Attention**
>
> This API is currently available in C++ and Python.

```
class OEMolFunc2 : public OEOpt::OEFunc2, public OEMolFunc1
```

The *OEMolFunc2* is an abstract base class. This class defines the interface for stateful functions which operate on molecules and conformers, compute a function value, gradients, and second derivatives. *OEMolFunc2* combines *OEFunc2* with *OEMolFunc* interface. Classes derived from *OEMolFunc2* can be used with any optimizers.

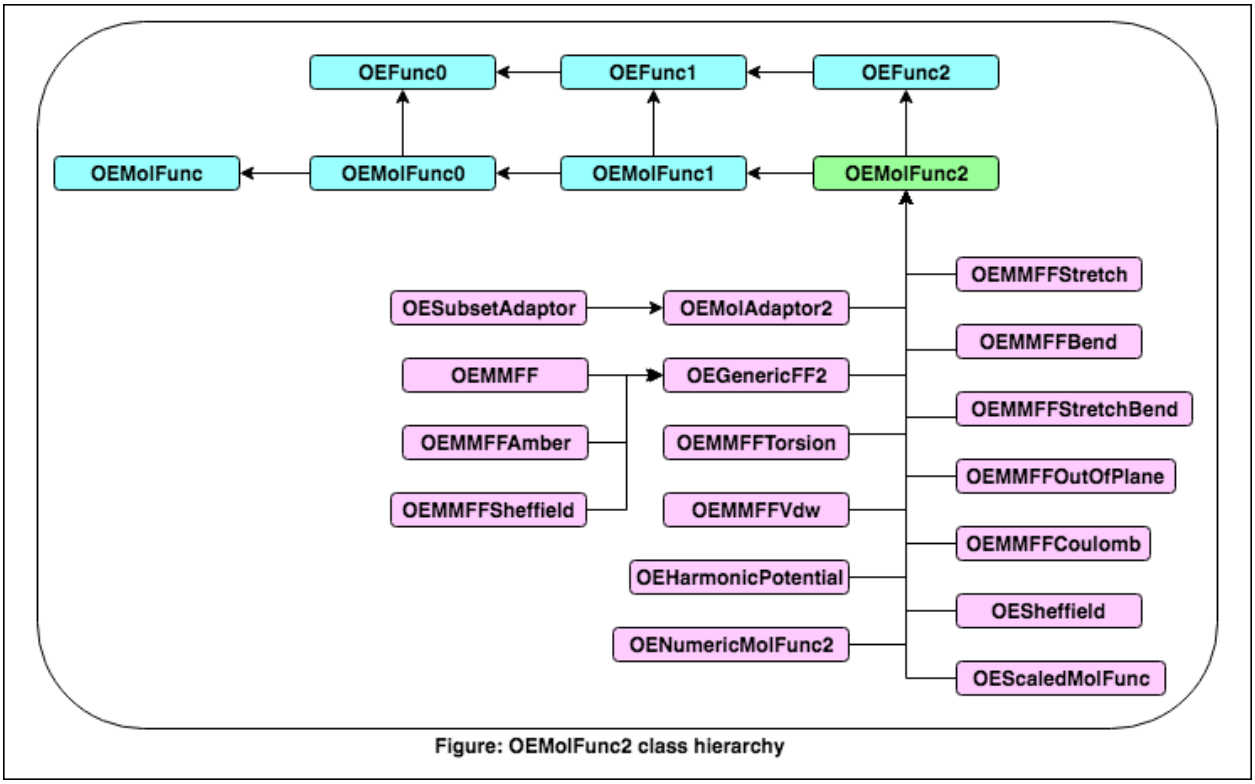


Figure: OEMolFunc2 class hierarchy

The following methods are publicly inherited from *OEFunc0*:

- *operator()*
- *GetFComponents*
- *NumVar*

The following methods are publicly inherited from *OEFunc1*:

- *operator()*

The following methods are publicly inherited from *OEFunc2*:

- *operator()*

The following methods are publicly inherited from *OEMolFunc*:

- *Clear*
- *ClearPredicates*
- *GetInteractions*
- *Set*
- *Setup*

The following classes derive from this class:

- *OEGenericFF2*
- *OESubsetAdaptor*
- *OENumericMolFunc2*
- *OEScaledMolFunc*
- *OEFFPotential*
- *OEStretchPotential*
- *OEBendPotential*
- *OETorsionPotential*
- *OEOutOfPlanePotential*
- *OENonBondPotBase*
- *OEVdwPotential*
- *OECoulombPotential*
- *OEInterMolFunc2*
- *OEInterNonBondPotBase*
- *OEInterNonBondPotential*
- *OEInterVdwPotential*
- *OEInterCoulombPotential*
- *OEMMFF*
- *OEMMFFStretchBend*
- *OESheffield*
- OEHermiteShapeFunc

3.2.24 OENonBondPotBase

Attention

This API is currently available in C++ and Python.

```
class OENonBondPotBase : public virtual OEMolPotential::OEFFPotential
```

The *OENonBondPotBase* is an abstract base class. This class defines a common base class for all intra-molecular non-bonded force field interaction potentials.

The following methods are publicly inherited from *OEFunc0*:

- *operator()*
- *GetFComponents*
- *NumVar*

The following methods are publicly inherited from *OEFunc1*:

- *operator()*

The following methods are publicly inherited from *OEFunc2*:

- *operator()*

The following methods are publicly inherited from *OEMolFunc*:

- *Clear*
- *ClearPredicates*
- *GetInteractions*
- *Setup*

The following methods are publicly inherited from *OEFFPotential*:

- *Set*

The *OENonBondPotBase* implements the following methods:

- *GetNonBondOptions*
- *Set*

The following classes derive from this class:

- *OENonBondPotential*
- *OEVdwPotential*
- *OECoulombPotential*

Constructors

```
OENonBondPotBase(const OEMolPotential::OEFFParams& fp,
                 const OENonBondIntcsOptions& options = OENonBondIntcsOptions())
OENonBondPotBase(const OENonBondPotBase&)
```

Constructor and copy constructor.

Assignment operator

```
OENonBondPotBase& operator=(const OENonBondPotBase&)
```

Assignment operator.

GetNonBondOptions

```
OENonBondIntcsOptions& GetNonBondOptions()
const OENonBondIntcsOptions& GetNonBondOptions() const
```

Returns *options* related to non-bonded interactions.

Set

```
bool Set(const OESystem::OEBinaryPredicate<OEChem::OEAtomBase, OEChem::OEAtomBase>& pred,
         unsigned int flag = OEFuncType::All)
```

This method can be used for interaction-level control of the non-bonded terms. Interactions between two atoms are only included if the test of the binary predicate with both atoms returns `true`. The *Set* method must be called before the *Setup* for it to be effective.

3.2.25 OENonBondPotential

> **Attention**
>
> This API is currently available in C++ and Python.

```
class OENonBondPotential : public virtual OENonBondPotBase
```

The *OENonBondPotential* class defines an interface for calculation of intra-molecular non-bonded (both VdW and Coulomb) force field interaction potentials.

The following methods are publicly inherited from *OEFunc0*:

- *operator()*
- *GetFComponents*
- *NumVar*

The following methods are publicly inherited from *OEFunc1*:

- *operator()*

The following methods are publicly inherited from *OEFunc2*:

- *operator()*

The following methods are publicly inherited from *OEMolFunc*:

- *Clear*
- *ClearPredicates*
- *GetInteractions*
- *Setup*

The following methods are publicly inherited from *OEFFPotential*:

- *Set*

The following methods are publicly inherited from *OENonBondPotBase*:

- *GetNonBondOptions*
- *Set*

Constructors

```
OENonBondPotential(const OEMolPotential::OEFFParams& fp,
                   const OENonBondIntcsOptions& options = OENonBondIntcsOptions())
OENonBondPotential(const OENonBondPotential&)
```

Constructor and copy constructor.

Assignment operator

```
OENonBondPotential& operator=(const OENonBondPotential&)
```

Assignment operator.

3.2.26 OENonBondIntcsOptions

> **Attention**
>
> This API is currently available in C++ and Python.

```
class OENonBondIntcsOptions : public OESystem::OEOptions
```

The *OENonBondIntcsOptions* is a class that defines common options for *OENonBondPotBase*.

The *OENonBondIntcsOptions* class defines the following public methods:

- *CreateCopy*
- *GetCoulombCutoff*
- *GetUseCoulomb*
- *GetUseVdW*
- *GetVdWCutoff*
- *SetUseCoulomb*
- *SetUseVdW*

Constructors

```
OENonBondIntcsOptions()
OENonBondIntcsOptions(const OENonBondIntcsOptions&)
```

Constructor and copy constructor.

Assignment operator

```
OENonBondIntcsOptions& operator=(const OENonBondIntcsOptions&)
```

Assignment operator.

CreateCopy

```
OEMolPotential::OENonBondIntcsOptions* CreateCopy() const
```

This method returns a pointer to a deep copy of the instance of *OENonBondIntcsOptions*.

GetCutoff

```
float d = GetCutoff()
```

See *SetCutoff* method.

GetUseCoulomb

```
bool GetUseCoulomb() const
```

See *SetUseCoulomb* method.

GetUseVdW

```
bool GetUseVdW() const
```

See *SetUseVdW* method.

SetCutoff

```
bool SetCutoff(const double)
```

Sets the **cutoff** to be used for non-bonded interactions (van der Waals and coulombic potentials). A cutoff value of *0.0* implies that a cutoff value should not be used at all. **Default: 0.0**

SetUseCoulomb

```
bool SetUseCoulomb(bool)
```

Sets **flag specifying if coulombic interactions** should be included in calculation. **Default: True**

SetUseVdW

```
bool SetUseVdW(const bool)
```

Sets **flag specifying if van der Waals interactions** should be included in calculation. **Default: True**

3.2.27 OENumericMolFunc2

Attention

This API is currently available in C++ and Python.

```
class OENumericMolFunc2 : public OEMolFunc2
```

The *OENumericMolFunc2* class converts a *OEMolFunc1* into a *OEMolFunc2* by providing numeric hessians.

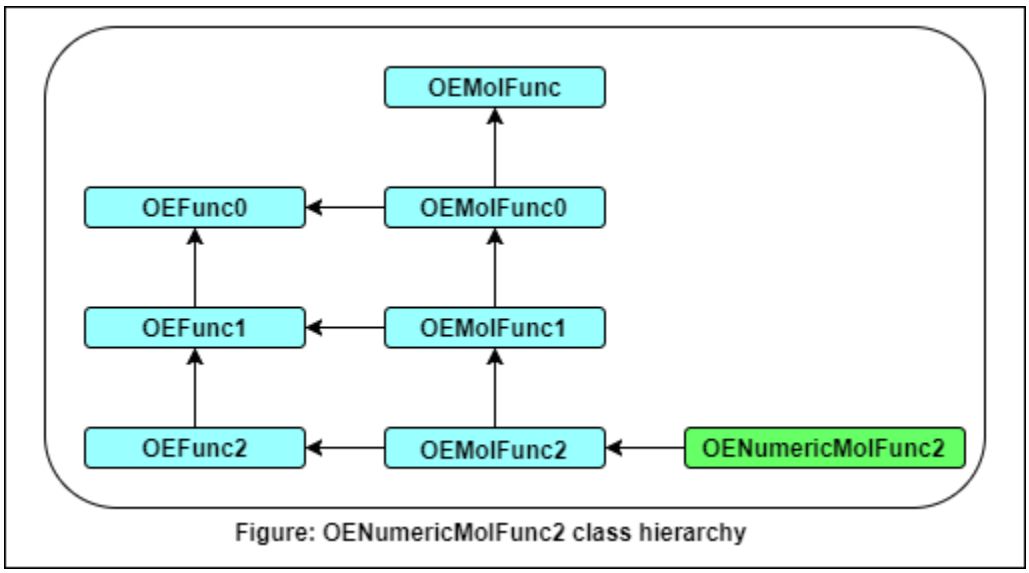


Figure: OENumericMolFunc2 class hierarchy

The following methods are publicly inherited from *OEFunc0*:

- *operator()*
- *GetFComponents*
- *NumVar*

The following methods are publicly inherited from *OEFunc1*:

- *operator()*

The following methods are publicly inherited from *OEFunc2*:

- *operator()*

The following methods are publicly inherited from *OEMolFunc*:

- *Clear*
- *ClearPredicates*
- *GetInteractions*
- *Setup*

The *OENumericMolFunc2* class defines the following public methods:

- *Set*

Constructors

```
OENumericMolFunc2(OEMolFunc1 &, bool own = false)
```

Default and copy constructors.

The molecule function used for function evaluation and to be converted must be provided as the first argument to the constructor. The second argument specifies the scaling factor. The third argument specifies whether the *OENumericMolFunc2* object takes ownership of the memory of the molecule function instance. By default that does not happen, so the *OENumericMolFunc2* destructor does not delete the molecule function instance. If ownership of the molecule function is transferred to the *OENumericMolFunc2* instance, the molecule function's delete operator will be called in the *OENumericMolFunc2* destructor.

Set

```
bool Set(const OESystem::OEUnaryPredicate<OEChem::OEBondBase>& pred,
         unsigned int functype)
bool Set(const OESystem::OEUnaryPredicate<OEChem::OEAtomBase>& pred,
         unsigned int functype)
bool Set(const OESystem::OEBinaryPredicate<OEChem::OEAtomBase, OEChem::OEAtomBase>& pred,
         unsigned int functype)
bool Set(const OESystem::OEBinaryPredicate<OEChem::OEBondBase, OEChem::OEBondBase>& pred,
         unsigned int functype)
```

Predicates are passed directly to the molecule function that is to be scaled.

3.2.28 OEOutOfPlaneParams

> **Attention**
>
> This API is currently available in C++ and Python.

```
class  OEOutOfPlaneParams : public OEMolPotential::OEInteractParams
```

The *OEOutOfPlaneParams* class defines generic parameters for out-of-plane term which might be used in any force field.

The following methods are publicly inherited from *OEInteractParams*:

- *Clear*
- *Get*
- *Set*

The *OETorsionParams* class defines the following public methods:

- *GetAtomIndex1*
- *GetAtomIndex2*
- *GetAtomIndex3*
- *GetAtomIndex4*
- *IsValid*

Constructors

```
OEOutOfPlaneParams(const OEChem::OEAtomBase* atom1,
                   const OEChem::OEAtomBase* atom2,
                   const OEChem::OEAtomBase* atom3,
                   const OEChem::OEAtomBase* atom4,
                   const unsigned type = OEFormatType::AMBER)
OEOutOfPlaneParams(const OEOutOfPlaneParams&)
```

Constructor and copy constructor. Constructs an *OEOutOfPlaneParams* object for for atoms defining improper torsion passed as arguments. Second atom (second argument) is an atom which might occure above or beyond the plane make by 3 remaining atoms.

GetAtomIndex1

```
unsigned GetAtomIndex1() const
```

Returns atom index of the first atom for an improper torsion for which the instance of *OEOutOfPlaneParams* was created.

GetAtomIndex2

```
unsigned GetAtomIndex2() const
```

Returns atom index of the second atom for an improper torsion for which the instance of *OEOutOfPlaneParams* was created.

GetAtomIndex3

```
unsigned GetAtomIndex3() const
```

Returns atom index of the third atom for an improper torsion for which the instance of *OEOutOfPlaneParams* was created.

GetAtomIndex4

```
unsigned GetAtomIndex4() const
```

Returns atom index of the forth atom for an improper torsion for which the instance of *OEOutOfPlaneParams* was created.

IsValid

```
bool IsValid(const OESystem::OEUnaryPredicate<OEChem::OEAtomBase>& pred) const
```

Method which checks if an instance of the *OEOutOfPlaneParams* class was created for the improper torsion to be included in force field calculation. Method provides a predicate which should be used to test all 4 atoms used to make this instance.

3.2.29 OEOutOfPlanePotential

Attention

This API is currently available in C++ and Python.

```
class OEOutOfPlanePotential : public virtual OEMolPotential::OEFFPotential
```

The *OEOutOfPlanePotential* defines an interface for all out-of-plane force field interaction potentials.

The following methods are publicly inherited from *OEFunc0*:

- *operator()*
- *GetFComponents*
- *NumVar*

The following methods are publicly inherited from *OEFunc1*:

- *operator()*

The following methods are publicly inherited from *OEFunc2*:

- *operator()*

The following methods are publicly inherited from *OEMolFunc*:

- *Clear*
- *ClearPredicates*
- *GetInteractions*
- *Setup*

The following methods are publicly inherited from *OEFFPotential*:

- *Set*

Constructors

```
OEOutOfPlanePotential(const OEFFParams&)
OEOutOfPlanePotential(const OEOutOfPlanePotential&)
```

Constructor and copy constructor.

Assignment operator

```
OEOutOfPlanePotential& operator=(const OEOutOfPlanePotential&)
```

Assignment operator.

3.2.30 OEQuatAdaptor

Attention

This API is currently available in C++ and Python.

```
class OEQuatAdaptor : public OEMolAdaptor
```

The *OEQuatAdaptor* class provides the ability to rotate and translate a molecule as single rigid unit or solid body in the presence of an external field. Rotations are represented with quaternion variables.

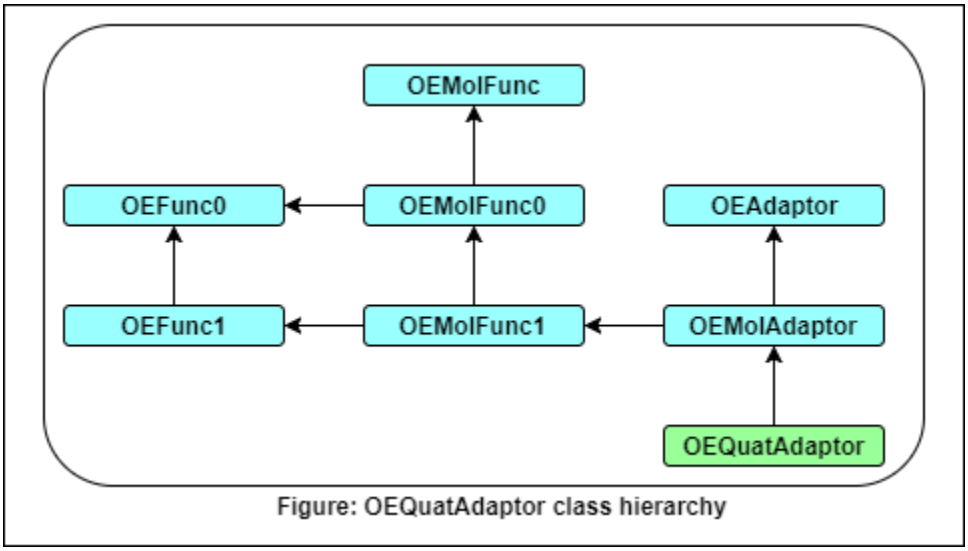


Figure: OEQuatAdaptor class hierarchy

The following methods are publicly inherited from *OEAdaptor*:

- *AdaptVar*
- *GetVar*

The following methods are publicly inherited from *OEFunc0*:

- *operator()*
- *GetFComponents*
- *NumVar*

The following methods are publicly inherited from *OEFunc1*:

- *operator()*

The following methods are publicly inherited from *OEMolFunc*:

- *Clear*
- *ClearPredicates*
- *GetInteractions*
- *Setup*

The following methods are publicly inherited from *OEMolAdaptor*:

- *SetMolFunc*

The *OEQuatAdaptor* class defines the following public methods:

- *AdaptGrad*
- *IsRedundant*

Constructors

```
OEQuatAdaptor(OEMolFunc1 &, bool own = false,
              bool excludeInteract = true)
```

Default and copy constructors.

Constructs an *OEQuatAdaptor* instance. The molecule function used for function evaluation must be provided as the first argument to the constructor. The second argument specifies whether the *OEQuatAdaptor* object takes ownership of the memory of the molecule function instance. By default that does not happen, so the *OEQuatAdaptor* destructor does not delete the molecule function instance. If ownership of the molecule function is transferred to the *OEQuatAdaptor* instance, the molecule function's delete operator will be called in the *OEQuatAdaptor* destructor. The third argument controls the inclusion of invariant molecule function interactions. By default, interactions which remain constant during the solid-body optimization (all intramolecular interactions) are excluded from calculation. If the third calling argument is `false` then all interactions will be computed during every function evaluation.

AdaptGrad

```
bool AdaptGrad(double* qg, const double* g, const double* quat,
               const double* coords) const
```

Transforms a set of the input Cartesian gradients (second argument) into rotation and translation gradients which are copied to the first argument array. Third and fourth arguments are the arrays of quaternion plus translation vector and Cartesian coordinates respectively, used for the transformation.

IsRedundant

```
bool IsRedundant(const double* x, const double* y,
                 const double tolerance = 0.1) const
```

Compares two sets of quaternions-translational vectors coordinates. If points `x` and `y` represent the same point in Cartesian space within given tolerance, method returns `true`.

3.2.31 OEScaledMolFunc

Attention

This API is currently available in C++ and Python.

```
class OEScaledMolFunc : public OEMolFunc2
```

The *OEScaledMolFunc* class allows scaling energies, gradients and hessians of a molecule function, with a specified constant.

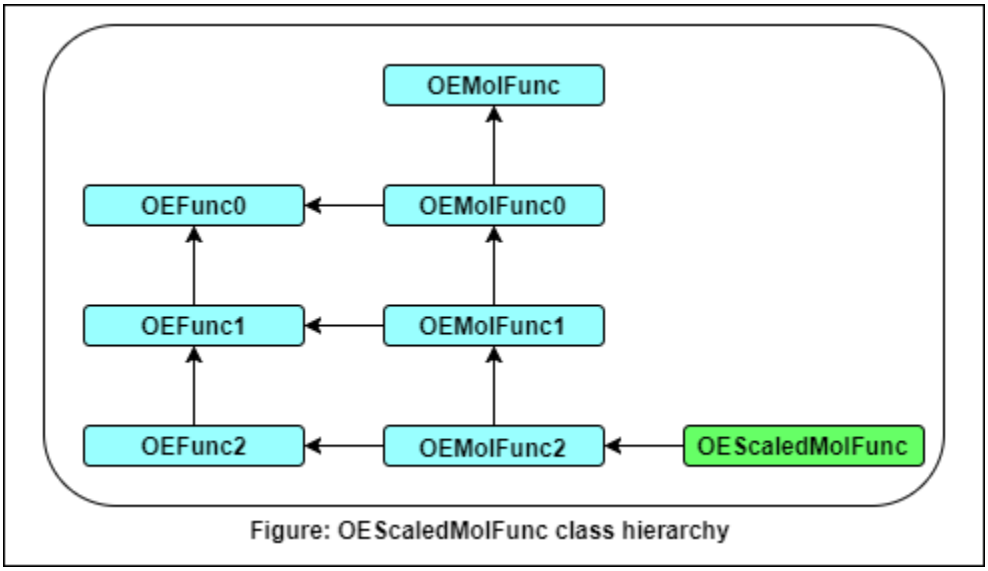


Figure: OEScaledMolFunc class hierarchy

The following methods are publicly inherited from *OEFunc0*:

- *operator()*
- *GetFComponents*
- *NumVar*

The following methods are publicly inherited from *OEFunc1*:

- *operator()*

The following methods are publicly inherited from *OEFunc2*:

- *operator()*

The following methods are publicly inherited from *OEMolFunc*:

- *Clear*
- *ClearPredicates*
- *GetInteractions*
- *Setup*

The *OEScaledMolFunc* class defines the following public methods:

- *Set*

Constructors

```
OEScaledMolFunc(OEMolFunc1 &, const double factor, bool own = false)
OEScaledMolFunc(OEMolFunc2 &, const double factor, bool own = false)
```

Default and copy constructors.

The molecule function used for function evaluation and to be scaled must be provided as the first argument to the constructor. The second argument specifies the scaling factor. The third argument specifies whether the *OEScaledMolFunc* object takes ownership of the memory of the molecule function instance. By default that does not happen, so the *OEScaledMolFunc* destructor does not delete the molecule function instance. If ownership of the molecule function is transferred to the *OEScaledMolFunc* instance, the molecule function's delete operator will be called in the *OEScaledMolFunc* destructor.

Set

```
bool Set(const OESystem::OEUnaryPredicate<OEChem::OEBondBase>& pred,
         unsigned int functype)
bool Set(const OESystem::OEUnaryPredicate<OEChem::OEAtomBase>& pred,
         unsigned int functype)
bool Set(const OESystem::OEBinaryPredicate<OEChem::OEAtomBase, OEChem::OEAtomBase>& pred,
         unsigned int functype)
bool Set(const OESystem::OEBinaryPredicate<OEChem::OEBondBase, OEChem::OEBondBase>& pred,
         unsigned int functype)
```

Predicates are passed directly to the molecule function that is to be scaled.

3.2.32 OEStretchParams

> **Attention**
>
> This API is currently available in C++ and Python.

```
class  OEStretchParams : public OEMolPotential::OEInteractParams
```

The *OEStretchParams* class defines generic bond stretch parameters used in any force field which has bond stretch term.

The following methods are publicly inherited from *OEInteractParams*:

- *Clear*
- *Get*
- *Set*

The *OEStretchParams* class defines the following public methods:

- *GetAtomIndex1*
- *GetAtomIndex2*
- *IsValid*
- *IsBond*

Constructors

```
OEStretchParams(const OEChem::OEBondBase* bond,
                const unsigned type = OEFormatType::AMBER)
OEStretchParams(const OEStretchParams&)
```

Constructor and copy constructor. Constructs an *OEStretchParams* object for the bond passed as argument.

GetAtomIndex1

```
unsigned GetAtomIndex1() const
```

Returns atom index of the first atom making a bond for which the instance of *OEStretchParams* was created.

GetAtomIndex2

```
unsigned GetAtomIndex2() const
```

Returns atom index of the second atom making a bond for which the instance of *OEStretchParams* was created.

IsValid

```
bool IsValid(const OESystem::OEUnaryPredicate<OEChem::OEAtomBase>& pred) const
bool IsValid(const OESystem::OEUnaryPredicate<OEChem::OEBondBase>& pred) const
```

Methods which check if an instance of the *OEStretchParams* class was created for a bond to be included in force field calculation. The first method provides a predicate which tests each of the 2 atoms belonging to a bond for which the above instance was created, while the second method provides a predicate which tests a particular bond.

IsBond

```
bool IsBond(const OEChem::OEBondBase* bond) const
```

Used to check if the bond passed as an argument belongs to the current *OEStretchParams* instance.

3.2.33 OEStretchPotential

> **Attention**
>
> This API is currently available in C++ and Python.

```
class OEStretchPotential : public virtual OEMolPotential::OEFFPotential
```

The *OEStretchPotential* defines an interface all bond stretching force field interaction potentials.

The following methods are publicly inherited from *OEFunc0*:

- *operator()*
- *GetFComponents*
- *NumVar*

The following methods are publicly inherited from *OEFunc1*:

- *operator()*

The following methods are publicly inherited from *OEFunc2*:

- *operator()*

The following methods are publicly inherited from *OEMolFunc*:

- *Clear*
- *ClearPredicates*
- *GetInteractions*
- *Setup*

The following methods are publicly inherited from *OEFFPotential*:

- *Set*

The *OEStretchPotential* implements the following methods:

- *Set*

Constructors

```
OEStretchPotential(const OEFFParams&)
OEStretchPotential(const OEStretchPotential&)
```

Constructor and copy constructor.

Assignment operator

```
OEStretchPotential& operator=(const OEStretchPotential&)
```

Assignment operator.

Set

```
bool Set(const OESystem::OEUnaryPredicate<OEChem::OEBondBase>&,
         unsigned int flag = OEFuncType::Stretch) override
```

This method can be used for interaction-level control of the bond stretching terms. Interactions between two atoms are only included if the test of the unary bond predicate returns `true`. The *Set* method must be called before the *Setup* for it to be effective.

3.2.34 OESubsetAdaptor

> **Attention**
>
> This API is currently available in C++ and Python.

```
class OESubsetAdaptor : public OEMolAdaptor2
```

The *OESubsetAdaptor* class allows optimization of Cartesian coordinates for a subset of atoms in a molecule, while the remaining atoms are fixed. This is accomplished by subsetting the coordinates and gradients into the subsets which correspond to atoms being optimized. Additional facilities are provided for excluding invariant molecule function interactions, and controlling which subset of atoms are held fixed during an optimization.

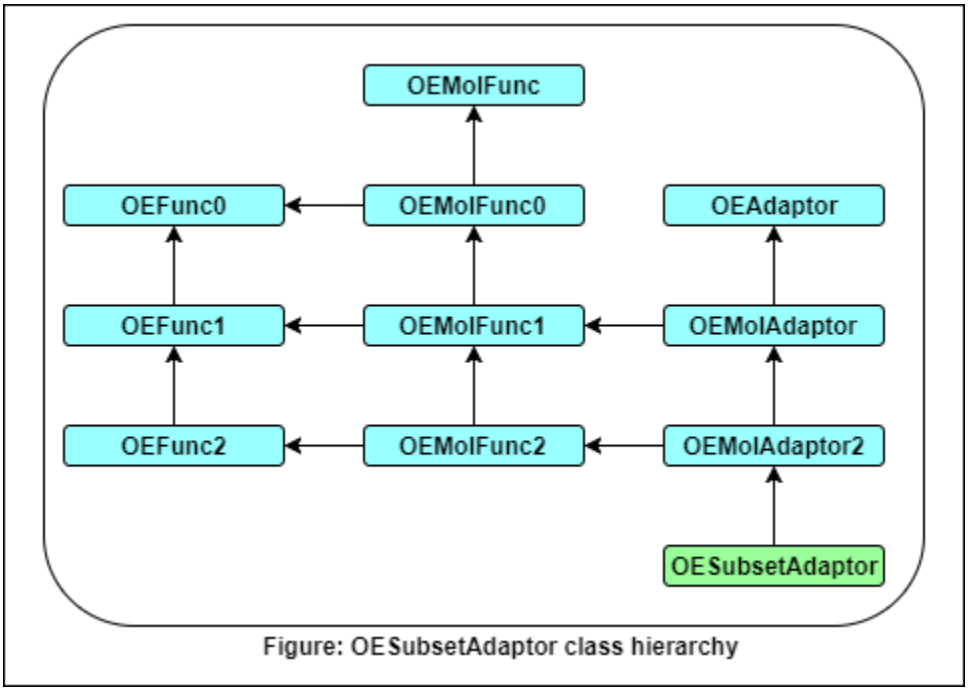


Figure: OESubsetAdaptor class hierarchy

The following methods are publicly inherited from *OEAdaptor*:

- *AdaptVar*
- *GetVar*

The following methods are publicly inherited from *OEFunc0*:

- *operator()*
- *GetFComponents*
- *NumVar*

The following methods are publicly inherited from *OEFunc1*:

- *operator()*

The following methods are publicly inherited from *OEFunc2*:

- *operator()*

The following methods are publicly inherited from *OEMolFunc*:

- *Clear*
- *ClearPredicates*
- *GetInteractions*
- *Setup*

The following methods are publicly inherited from *OEMolAdaptor*:

- *SetMolFunc*

The following methods are publicly inherited from *OEMolAdaptor2*:

- *SetMolFunc*

The *OESubsetAdaptor* class defines the following public methods:

- *AdaptGrad*
- *AdaptHessian*
- *Set*

Constructors

```
OESubsetAdaptor(OEMolFunc1 &, bool own = false,
                bool excludeInteract = true)
OESubsetAdaptor(OEMolFunc2 &, bool own = false,
                bool excludeInteract = true)
```

Default and copy constructors.

The molecule function used for function evaluation must be provided as the first argument to the constructor. The second argument specifies whether the *OESubsetAdaptor* object takes ownership of the memory of the molecule function instance. By default that does not happen, so the *OESubsetAdaptor* destructor does not delete the molecule function instance. If ownership of the molecule function is transferred to the *OESubsetAdaptor* instance, the molecule function's delete operator will be called in the *OESubsetAdaptor* destructor. The third argument controls the inclusion of invariant molecule function interactions. By default, interactions which remain constant during the optimization are excluded from calculation. Interactions composed entirely of atoms held fixed during the optimization will not change during the optimization. Excluding invariant interactions does not affect the optimization, however, the total energy calculated for a set of coordinates will be altered by the exclusion. If the third calling argument is `false` then all interactions will be computed during every function evaluation.

AdaptGrad

```
bool AdaptGrad(double* ga, const double* g) const
```

Takes a set of gradients from a molecule function evaluation as the second argument, and subsets the gradients into the array passed as the first argument.

AdaptHessian

```
bool AdaptHessian(double* ha, const double* h) const
```

Takes a set of hessians from a molecule function evaluation as the second argument, and subsets the hessians into the array passed as the first argument.

Set

```
bool Set(const OESystem::OEUnaryPredicate<OEChem::OEAtomBase>&,
         unsigned int functype = OEFuncType::SubsetAdapt);)
```

This method assigns an atom predicate which is used to specify atoms which are to be held fixed in their starting positions during an optimization. The *Set* method must be called before the *Setup* for it to be effective.

3.2.35 OETorAdaptor

Attention

This API is currently available in C++ and Python.

```
class OETorAdaptor : public OEMolAdaptor
```

The *OETorAdaptor* class provides the ability to optimize torsion angles of a molecule while holding bond lengths and bond angles fixed. *OEMolFunc1* derived classes typically require Cartesian coordinates for function evaluation. *OETorAdaptor* transforms a set of torsions into Cartesian coordinates, passes the resultant coordinates to the internally stored *OEMolFunc1* type function, and transforms the Cartesian gradients back into torsion gradients. Additional facilities are provided for excluding invariant molecule function interactions, controlling the definition of a rotatable bond, and fixing atomic positions during coordinate transformation.

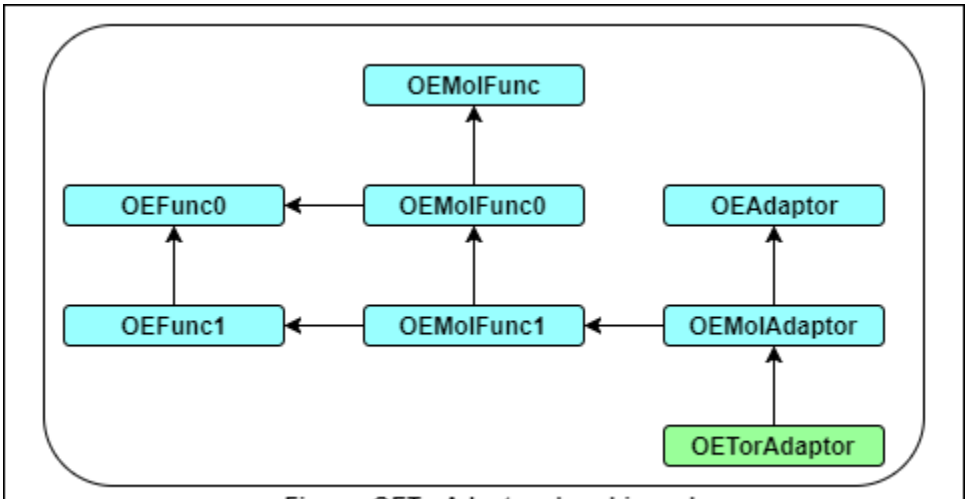


Figure: OETorAdaptor class hierarchy

The following methods are publicly inherited from *OEAdaptor*:

- *AdaptVar*
- *GetVar*

The following methods are publicly inherited from *OEFunc0*:

- *operator()*
- *GetFComponents*
- *NumVar*

The following methods are publicly inherited from *OEFunc1*:

- *operator()*

The following methods are publicly inherited from *OEMolFunc*:

- *Clear*
- *ClearPredicates*
- *GetInteractions*
- *Setup*

The following methods are publicly inherited from *OEMolAdaptor*:

- *SetMolFunc*

The *OETorAdaptor* class defines the following public methods:

- *AdaptGrad*
- *Set*

Constructors

```
OETorAdaptor(OEMolFunc1 &, bool own = false,
             bool excludeInteract = true)
```

Default and copy constructors.

Constructs an *OETorAdaptor* instance. The molecule function used for function evaluation must be provided as the first argument to the constructor. The second argument specifies whether the *OETorAdaptor* object takes ownership of the memory of the molecule function instance. By default the *OETorAdaptor* instance does not take ownership of the molecule function, so the *OETorAdaptor* destructor does not delete the molecule function instance. If ownership of the molecule function is transferred to the *OETorAdaptor* instance, the molecule function's delete operator will be called in the *OETorAdaptor* destructor. The third argument controls the inclusion of invariant molecule function interactions. By default, interactions which remain constant during the torsion minimization are excluded from calculation (for example bond stretch interactions). If the third calling argument is `false` then all interactions will be computed during every function evaluation.

AdaptGrad

```
bool AdaptGrad(double* torGrad, const double* grads) const
```

Takes a set of Cartesian gradients of a conformer (second argument), and transforms them into angular torsion gradients (first argument).

Set

```
bool Set(const OESystem::OEUnaryPredicate<OEChem::OEAtomBase> &p,
         unsigned int functype = OEFuncType::TorAdapt)
bool Set(const OESystem::OEUnaryPredicate<OEChem::OEBondBase> &rot,
         unsigned int functype = OEFuncType::TorAdapt)
```

The first method assigns an atom predicate which is used to specify atoms which are to be held fixed in their starting positions during an optimization. The second method assigns a bond predicate which is used to determine which bonds of a molecule are to be rotated during an optimization. Both the *Set* methods must be called before the *Setup* for it to be effective.

3.2.36 OETorQuatAdaptor

> **Attention**
>
> This API is currently available in C++ and Python.

```
class OETorQuatAdaptor : public OEMolAdaptor
```

The *OETorQuatAdaptor* class provides the ability to optimize a molecule in coordinate space by a combination of rotation, translation and torsion changes. This class combines the functionality of both *OEQuatAdaptor* and *OETorAdaptor* classes. Its main application is optimization of ligands in the potential generated by protein-receptor, in the case when ligand's bond lengths and bond angles are chosen to be fixed.

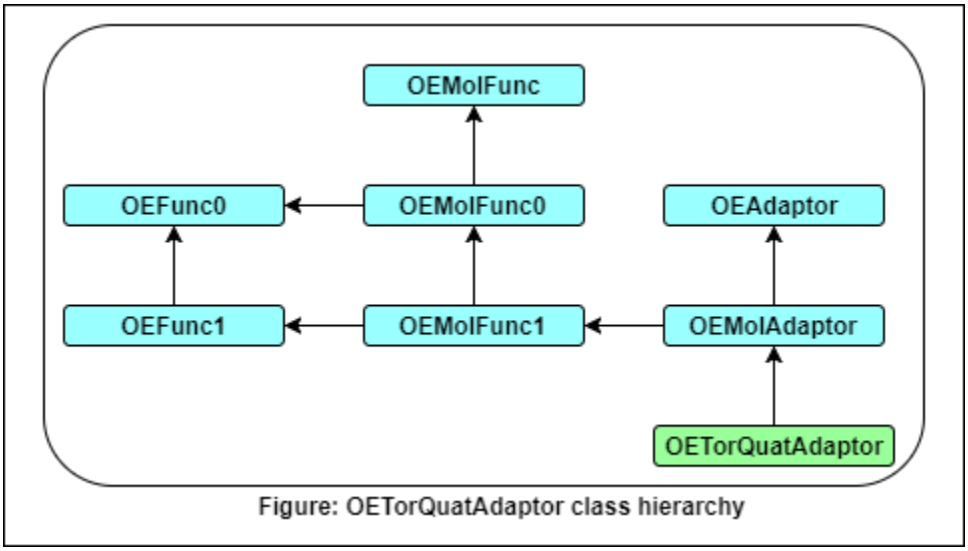


Figure: OETorQuatAdaptor class hierarchy

The following methods are publicly inherited from *OEAdaptor*:

- *AdaptVar*
- *GetVar*

The following methods are publicly inherited from *OEFunc0*:

- *operator()*
- *GetFComponents*
- *NumVar*

The following methods are publicly inherited from *OEFunc1*:

- *operator()*

The following methods are publicly inherited from *OEMolFunc*:

- *Clear*
- *ClearPredicates*
- *GetInteractions*
- *Setup*

The following methods are publicly inherited from *OEMolAdaptor*:

- *SetMolFunc*

The *OETorQuatAdaptor* class defines the following public methods:

- *AdaptGrad*
- *Set*

Constructors

```
OETorQuatAdaptor(OEMolFunc1 &, bool own = false,
                 bool excludeInteract = true)
```

Default and copy constructors.

The molecule function used for function evaluation must be provided as the first argument to the constructor. The second argument specifies whether the *OETorQuatAdaptor* object takes ownership of the memory of the molecule function instance. By default the *OETorQuatAdaptor* instance does not take ownership of the molecule function, so the *OETorQuatAdaptor* destructor does not delete the molecule function instance. If ownership of the molecule function is transferred to the *OETorQuatAdaptor* instance, the molecule function's delete operator will be called in the *OETorQuatAdaptor* destructor. The third argument controls the inclusion of invariant molecule function interactions. By default, interactions which remain constant during the torsion minimization are excluded from calculation (for example bond stretch interactions). If the third calling argument is `false` then all interactions will be computed during every function evaluation.

AdaptGrad

```
bool AdaptGrad(double *, const double *, const double *) const
```

Transforms a set of the input Cartesian gradients passed to the method as a second argument, into torsional-rotational-translational gradients which are copied to the first argument array. The third argument to the method is an array of the current quaternion and translational vector.

Set

```
bool Set(const OESystem::OEUnaryPredicate<OEChem::OEAtomBase> &p,
         unsigned int f = OEFuncType::TorAdapt)
bool Set(const OESystem::OEUnaryPredicate<OEChem::OEBondBase> &p,
         unsigned int f = OEFuncType::TorAdapt)
```

The first method assigns an atom predicate which is used to specify atoms which are to be held fixed in their starting positions during an optimization. The second method assigns a bond predicate which is used to determine which bonds of a molecule are to be rotated during an optimization. Both the *Set* methods must be called before the *Setup* for it to be effective.

3.2.37 OETorsionParams

> **Attention**
>
> This API is currently available in C++ and Python.

```
class  OETorsionParams : public OEMolPotential::OEInteractParams
```

The *OETorsionParams* class defines generic torsion parameters used in any force field which has torsion term.

The following methods are publicly inherited from *OEInteractParams*:

- *Clear*
- *Get*
- *Get*

- *Set*
- *Set*

The *OETorsionParams* class defines the following public methods:

- *GetAtomIndex1*
- *GetAtomIndex2*
- *GetAtomIndex3*
- *GetAtomIndex4*
- *IsValid*
- *IsTorsion*

Constructors

```
OETorsionParams(const OEChem::OEBondBase* bond, const OEChem::OEAtomBase* atom1,
const OEChem::OEAtomBase* atom2, const unsigned type = OEFormatType::AMBER)
OETorsionParams(const OETorsionParams&)
```

Constructor and copy constructor. Constructs an *OETorsionParams* object for a middle bond passed as the first argument and two terminal atoms (second and third arguments).

GetAtomIndex1

```
unsigned GetAtomIndex1() const
```

Returns atom index of the first atom for a torsion for which the instance of *OETorsionParams* was created.

GetAtomIndex2

```
unsigned GetAtomIndex2() const
```

Returns atom index of the second atom for a torsion for which the instance of *OETorsionParams* was created.

GetAtomIndex3

```
unsigned GetAtomIndex3() const
```

Returns atom index of the third atom for a torsion for which the instance of *OETorsionParams* was created.

GetAtomIndex4

```
unsigned GetAtomIndex4() const
```

Returns atom index of the forth atom for a torsion for which the instance of *OETorsionParams* was created.

IsValid

```
bool IsValid() const
bool IsValid(const OESystem::OEUnaryPredicate<OEChem::OEAtomBase>& pred) const
bool IsValid(const OESystem::OEUnaryPredicate<OEChem::OEBondBase>& pred) const
```

Methods which check if an instance of the *OETorsionParams* class was created for the torsion to be included in force field calculation. First method is checking if the above instance was correctly constructed, while two remaining methods provide predicates which test if atoms (second method) and a middle bond (third method) belong to that instance.

IsTorsion

```
bool IsTorsion(const OEChem::OEAtomBase* atom1, const OEChem::OEAtomBase* atom2,
const OEChem::OEAtomBase* atom3, const OEChem::OEAtomBase* atom4) const
```

Used to check if 4 atoms passed as arguments belong to the current *OETorsionParams* instance.

3.2.38 OETorsionPotential

> **Attention**
>
> This API is currently available in C++ and Python.

```
class OETorsionPotential : public virtual OEMolPotential::OEFFPotential
```

The *OETorsionPotential* defines an interface for all torsional force field interaction potentials.

The following methods are publicly inherited from *OEFunc0*:

- *operator()*
- *GetFComponents*
- *NumVar*

The following methods are publicly inherited from *OEFunc1*:

- *operator()*

The following methods are publicly inherited from *OEFunc2*:

- *operator()*

The following methods are publicly inherited from *OEMolFunc*:

- *Clear*
- *ClearPredicates*
- *GetInteractions*
- *Setup*

The following methods are publicly inherited from *OEFFPotential*:

- *Set*

The *OETorsionPotential* implements the following methods:

- *Set*

Constructors

```
OETorsionPotential(const OEFFParams&)
OETorsionPotential(const OETorsionPotential&)
```

Constructor and copy constructor.

Assignment operator

```
OETorsionPotential& operator=(const OETorsionPotential&)
```

Assignment operator.

Set

```
bool Set(const OESystem::OEBinaryPredicate<OEChem::OEBondBase, OEChem::OEBondBase>&)
```

This method provides interaction-level control of the torsion terms. The method tests the central bond of a torsion angle. If such a bond returns *true* from the predicate test then all torsions that cross the bond are included in the list of interactions to be calculated. The *Set* method must be called before the *Setup* for it to be effective.

3.2.39 OEVdwParams

> **Attention**
>
> This API is currently available in C++ and Python.

```
class  OEVdwParams : public OEMolPotential::OEInteractParams
```

The *OEVdwParams* class defines generic vdw parameters used in any force field which has Lennard-Jones 6-12 (or similar 7-14) non-bonded term.

The following methods are publicly inherited from *OEInteractParams*:

- *Clear*
- *Get*
- *Set*

The *OEVdwParams* class defines the following public methods:

- *GetAtomIndex1*
- *GetAtomIndex2*
- *GetAtom1*
- *GetAtom2*
- *IsValid*
- *IsPair*

Constructors

```
OEVdwParams(const OEChem::OEAtomBase* atom1,
            const OEChem::OEAtomBase* atom2,
            const unsigned type = OEFormatType::AMBER)
OEVdwParams& operator=(const OEVdwParams&)
```

Constructor and copy constructor. Constructs an *OEVdwParams* object for a pair of atoms passed as arguments.

GetAtomIndex1

```
unsigned GetAtomIndex1() const
```

Returns atom index of the first atom from non-bonded interacting pair for which the instance of *OEVdwParams* was created.

GetAtomIndex2

```
unsigned GetAtomIndex2() const
```

Returns atom index of the second atom from non-bonded interacting pair for which the instance of *OEVdwParams* was created.

GetAtom1

```
const OEChem::OEAtomBase* GetAtom1() const
```

Returns the first atom from non-bonded interacting pair for which the instance of *OEVdwParams* was created.

GetAtom2

```
const OEChem::OEAtomBase* GetAtom2() const
```

Returns the second atom from non-bonded interacting pair for which the instance of *OEVdwParams* was created.

IsValid

```
bool IsValid(const OESystem::OEUnaryPredicate<OEChem::OEAtomBase>& pred) const
bool IsValid(const OESystem::OEBinaryPredicate<OEChem::OEAtomBase, OEChem::OEAtomBase>&␣
↪pred) const
```

Methods which check if an instance of the *OEVdwParams* class was created for a pair of non-bonded atoms to be included in force field calculation. The first method provides a predicate which is used to test individually each of the two atoms from the non-boned pair, while the second method provides a predicate which tests a non-bonded pair of atoms in a single test.

IsPair

```
bool IsPair(const OEChem::OEAtomBase* atom1,
            const OEChem::OEAtomBase* atom2) const
```

Used to check if atoms passed as arguments belong to the current *OEStretchParams* instance.

3.2.40 OEVdwPotential

> **Attention**
>
> This API is currently available in C++ and Python.

```
class OEVdwPotential : public virtual OENonBondPotBase
```

The *OEVdwPotential* class defines an interface for calculation of intra-molecular VdW force field interaction potentials.

The following methods are publicly inherited from *OEFunc0*:

- *operator()*
- *GetFComponents*
- *NumVar*

The following methods are publicly inherited from *OEFunc1*:

- *operator()*

The following methods are publicly inherited from *OEFunc2*:

- *operator()*

The following methods are publicly inherited from *OEMolFunc*:

- *Clear*
- *ClearPredicates*
- *GetInteractions*
- *Setup*

The following methods are publicly inherited from *OEFFPotential*:

- *Set*

The following methods are publicly inherited from *OENonBondPotBase*:

- *GetNonBondOptions*
- *Set*

Constructors

```
OEVdwPotential(const OEMolPotential::OEFFParams& fp,
               const double rcut = 0.0)
OEVdwPotential(const OEVdwPotential&)
```

Constructor and copy constructor.

Assignment operator

```
OEVdwPotential& operator=(const OEVdwPotential&)
```

Assignment operator.

3.3 OEMolPotential Constants

3.3.1 OEFFParam

> **Attention**
>
> This API is currently available in C++ and Python.

This namespace contains constants used to define integer constants, associated with *OEInteractParams*, that are used to retrieve force field parameters.

r0

Equilibrium bond length.

EqDist

Equilibrium bond length.

EqDistance

Equilibrium bond length.

StretchConst

Bond stretching constant.

k

Bond stretching constant.

rij

Radius parameter in van der Waals potential.

rij

Square of radius parameter in van der Waals potential.

ContactDist

Radius parameter in van der Waals potential.

Epsilon

Energy parameter in van der Waals potential.

eps

Energy parameter in van der Waals potential.

a0

Equilibrium bending angle.

EqAngle

Equilibrium bending angle.

BendConst

Angle bending constant.

ka

Angle bending constant.

kb1

First constant term in the stretch-bend potential.

StretchBendConst1

First constant term in the stretch-bend potential.

StretchBendConst2

Second constant term in the stretch-bend potential.

kb2

Second constant term in the stretch-bend potential.

r1

First distance term in the stretch-bend potential.

r2

Second distance term in the stretch-bend potential.

OutOfPlaneConst

Out-of-plane potential constant.

OOP

Out-of-plane potential constant.

ko

Out-of-plane potential constant.

V1

First constant term in torsion potential.

V2

Second constant term in torsion potential.

V3

Third constant term in torsion potential.

v1

First constant term in torsion potential.

v2

Second constant term in torsion potential.

v3

Third constant term in torsion potential.

p1

Periodicity of improper torsion (used in SMIRNOFF)

ph1

Phase of improper torsion (used in SMIRNOFF)

ntt

Number of terms in torsion Fourier expansion

listKv

Identifies vector which stores torsion barrier heights

listDv

Identifies vector which stores a set of torsion degeneracies

listPv

Identifies vector which stores a set of torsion periodicities

listPhv

Identifies vector which stores a set of torsion phases

3.3.2 OEVarType

> **Attention**
>
> This API is currently available in C++ and Python.

This namespace contains constants which define the type of molecular coordinates to be optimized.

SolidBody

Defines 6 degrees of freedom: 3 translations and 3 rotations. Because rotations are represented with quaternions, the total number of variables is 7.

Torsions

Torsional coordinates which are defined as hindered rotations around single bonds. Molecule has as many torsions as the number of rotatable bonds.

Cart

3N Cartesian coordinates where N is the number of atoms in the molecule.

TranRotTor

A combination of SolidBody and Torsions defined above, usually used for the optimization of a ligand inside the protein binding site.

3.3.3 OEFuncType

> **Attention**
>
> This API is currently available in C++ and Python.

This namespace contains constants which define the type of force field potential components which might be used separately or in combination for the energy estimation of a molecular system. Constants defined below are passed to the methods *OEMolFunc.Set* as a second parameter.

Undefined

Undefined potential type.

VdW

Defines vdW potential type represented by the *OEVdwPotential* class.

VdWRepel

Same as above but without long range attractive forces.

Coulomb

Defines Coulomb potential type represented by the *OECoulombPotential* class.

Bond

Defines bond stretching potential type represented by the *OEStretchPotential* class.

Stretch

Same as above

Bend

Defines angle bending potential type represented by the *OEBendPotential* class.

StretchBend

Defines angle-stretch coupling potential type represented by the *OEMMFFStretchBend* class.

Torsion

Defines torsion potential type represented by the *OETorsionPotential* class.

ImproperTorsion

Defines improper torsion potential type represented by the *OEOutOfPlanePotential* class.

OutOfPlane

Same as above.

All

Defines a combination of all 7 above components.

AllRepel

Same as above with a modified vdW component of the `OEFuncType.VdWRepel` type.

InterMolVdW

Defines intermolecular vdW potential type represented by the *OEInterVdwPotential* class.

InterMolCoulomb

Defines intermolecular Coulomb potential type represented by the *OEInterCoulombPotential* class.

TorAdapt

Defines torsion adaptor potential type represented by the *OETorAdaptor* class.

QuatAdapt

Defines quaternion adaptor potential type represented by the *OEQuatAdaptor* class.

SubsetAdapt

Defines subset adaptor potential type represented by the *OESubsetAdaptor* class.

SubsetAdaptor

Same as above.

InterAdapt

Defines internal coordinates adaptor potential type represented by the *OEInterAdaptor* class.

HarmonicPot

Defines harmonic potential type represented by the *OEHarmonicPotential* class.

Sheffield

Defines Sheffield solvation potential type represented by the *OESheffield* class.

3.4 OEConstr Classes

3.4.1 OEDihedralQuadPenalty

> **Attention**
>
> This API is currently available in C++ and Python.

```
class OEDihedralQuadPenalty : public OEMolPotential::OEMolFunc2
```

The *OEDihedralQuadPenalty* class defines harmonic restraints to dihedrals or torsions. The restraint is applied on the cosine of the angle, in the form of $K * (cos(\alpha) - cos(\alpha_0))^2$.

The following methods are publicly inherited from *OEFunc0*:

- *operator()*
- *GetFComponents*
- *NumVar*

The following methods are publicly inherited from *OEFunc1*:

- *operator()*

The following methods are publicly inherited from *OEFunc2*:

- *operator()*

The following methods are publicly inherited from *OEMolFunc*:

- *Clear*
- *ClearPredicates*
- *GetInteractions*
- *Setup*
- *SetupConf*

The *OEDihedralQuadPenalty* class defines the following public methods:

- *GetPenaltyConstant*
- *GetSpecifiedAngle*

Constructors

```
OEDihedralQuadPenalty(const OEChem::OEAtomBase& a1,
                      const OEChem::OEAtomBase& a2,
                      const OEChem::OEAtomBase& a3,
                      const OEChem::OEAtomBase& a4, double alpha0,
                      double fc = 1e4)
const OEChem::OEAtomBase& a1, const OEChem::OEAtomBase& a2, const OEChem::OEAtomBase& a3,
↪ const OEChem::OEAtomBase& a4, double alpha0, double K)
OEDihedralQuadPenalty(const OEDihedralQuadPenalty &)
```

Default and copy constructors. Arguments *a1-a4* correspond to the four atoms that define the dihedral to be restrained, *alpha0* corresponds to the specified angle, and *K* corresponds to the penalty force constant.

GetPenaltyConstant

```
double GetPenaltyConstant() const
```

Returns the current value of the penalty force constant.

GetSpecifiedAngle

```
double GetSpecifiedAngle() const
```

Returns the current value of the specified angle *alpha0*.

3.4.2 OEHarmonicPotential

> **Attention**
>
> This API is currently available in C++ and Python.

```
class OEHarmonicPotential : public OEMolPotential::OEMolFunc1
```

The *OEHarmonicPotential* class defines harmonic constraint interactions between atoms.

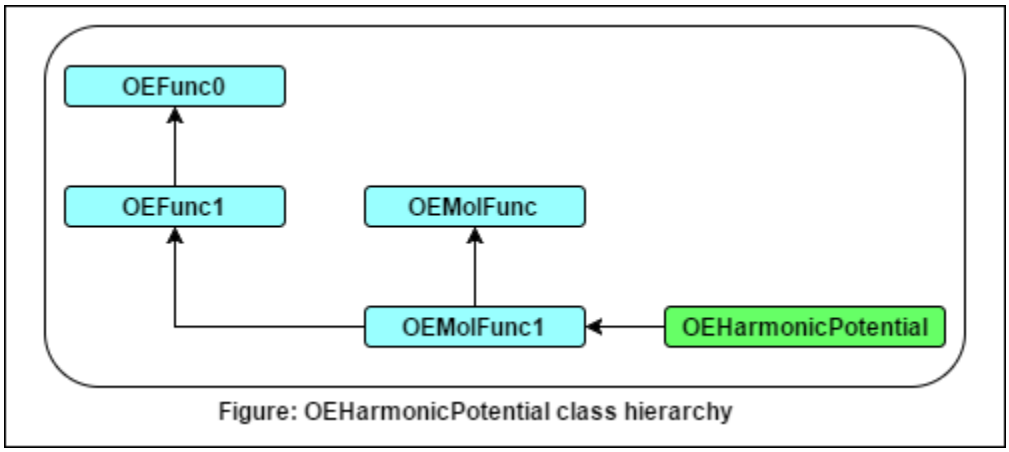


Figure: OEHarmonicPotential class hierarchy

The following methods are publicly inherited from *OEFunc0*:

- *operator()*
- *GetFComponents*
- *NumVar*

The following methods are publicly inherited from *OEFunc1*:

- *operator()*

The following methods are publicly inherited from *OEMolFunc*:

- *Clear*
- *ClearPredicates*
- *GetInteractions*
- *Setup*
- *SetupConf*

The *OEHarmonicPotential* class defines the following public methods:

- *Get_dc*
- *Get_kc*
- *Set*
- *Set_dc*
- *Set_kc*

Constructors

```
OEHarmonicPotential(double _kc, double _dc)
OEHarmonicPotential(const OEHarmonicPotential &)
```

Default and copy constructors. Parameters passed in the first constructor correspond to k and d in the harmonic potential $V = k(r - d)^2$.

Get_dc

```
double Get_dc() const
```

Returns the current value of parameter d (constraining distance) in constraining potential $V = k(r - d)^2$.

Get_kc

```
double Get_kc() const
```

Returns the current value of parameter k (force constant) in constraining potential $V = k(r - d)^2$.

Set

```
bool Set(const OESystem::OEUnaryPredicate<OEChem::OEAtomBase>&,
         unsigned int flag = OEFuncType::HarmonicPot);
```

The Set function can be used to set the unary atomic predicate used for the selective constraining of atoms. This function has to be used before Setup(const OEChem::OEMolBase&).

Set_dc

```
void Set_dc(double d)
```

Sets d parameter in $V = k(r - d)^2$.

Set_kc

```
void Set_kc(double k)
```

Sets k parameter in $V = k(r - d)^2$.

3.5 OEMolMMFF Classes

3.5.1 OEMMFF

> **Attention**
>
> This API is currently available in C++ and Python.

```
class OEMMFF : public OEMolPotential::OEGenericFF2
```

The *OEMMFF* class facilitates creation of an instance of a partial or full MMFF94 or MMFF94s force field. In addition, a force field instance can be extended by adding externally defined potential functions.

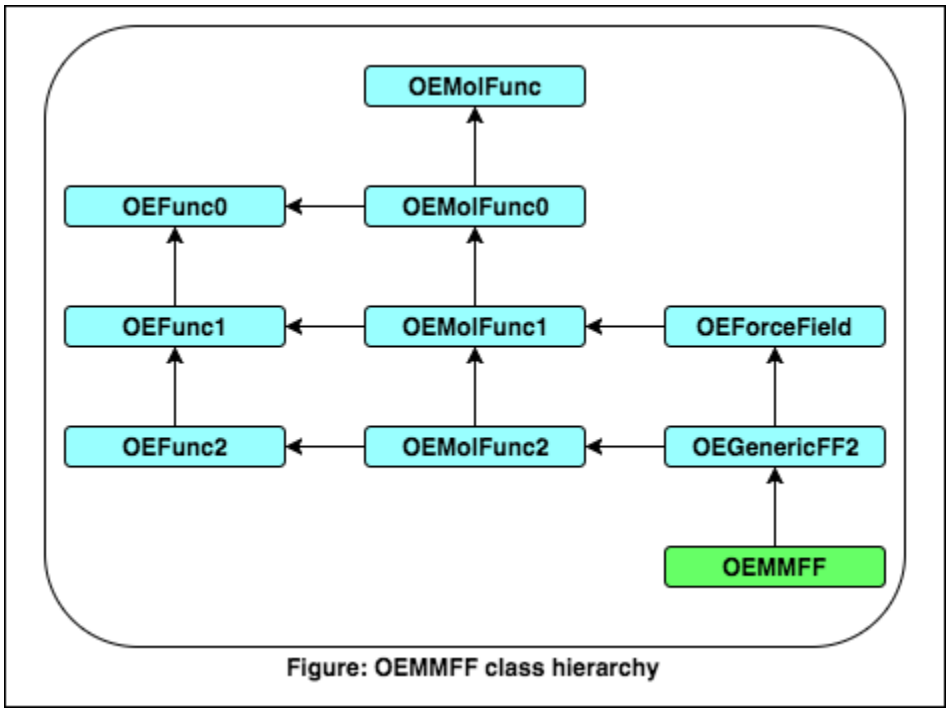


Figure: OEMMFF class hierarchy

The following methods are publicly inherited from *OEFunc0*:

- *operator()*
- *GetFComponents*
- *NumVar*

The following methods are publicly inherited from *OEFunc1*:

- *operator()*

The following methods are publicly inherited from *OEFunc2*:

- *operator()*

The following methods are publicly inherited from *OEMolFunc*:

- *Clear*
- *ClearPredicates*
- *GetInteractions*
- *Setup*

The following methods are publicly inherited from *OEForceField*:

- *GetTitle* and *SetTitle*
- *HasCoulomb*
- *HasInternalCharges*
- *PrepMol*
- *Set*

- *SetNonBondCutoff*

The following methods are publicly inherited from *OEGenericFF2*:

- *AddMolFunc*
- *GetComponents*
- *RemoveMolFunc*

Constructors

```
OEMMFF(const OEMMFFParams& params,
       const OEMolPotential::OENonBondIntcsOptions& opts =␣
↪OEMolPotential::OENonBondIntcsOptions())
OEMMFF(const OEMMFF&)
```

Default and copy constructors.

3.5.2 OEMMFF94sParams

> **Attention**
>
> This API is currently available in C++ and Python.

```
class OEMMFF94sParams : public OEMMFFParams
```

The *OEMMFF94sParams* class provides the ability to retrieve parameters necessary to perform energy and gradient calculations using the MMFF94s variant of the MMFF94 force field. Most force field terms are identical between MMFF94 and MMFF94s. The torsion and improper-torsion components of the MMFF94s have a different set of parameters which tend to planarize certain types of trigonal nitrogens.

The following methods are publicly inherited from *OEFFParams*:

- *CreateCopy*
- *GetCoulombParams*
- *GetStretchParams*
- *GetAtomTypeIndex*
- *GetEquilibriumAngle*
- *GetTorsionParams*
- *GetAtomTypeName*
- *GetEquilibriumBondLength*
- *GetVdwParams*
- *GetBendParams*
- *GetOutOfPlaneParams*
- *GetVdwRadius*
- *GetContactDistance*
- *GetStretchBendParams*

- *PrepMol*
- *GetVdwParams*
- *GetStretchParams*
- *GetBendParams*
- *GetTorsionParams*
- *GetOutOfPlaneParams*
- *HasVdwParams*
- *HasStretchParams*
- *HasBendParams*
- *HasTorsionParams*
- *GetTitle*

Constructors

```
OEMMFF94sParams()
```

Default and copy constructors.

3.5.3 OEMMFFOutOfPlane

> **Attention**
>
> This API is currently available in C++ and Python.

```
class OEMMFFOutOfPlane : public OEMolPotential::OEMolFunc2
```

The *OEMMFFOutOfPlane* class defines out-of-plane (improper torsion) interaction function using the functional forms used in MMFF94 force field. The function can be initialized with parameters defined in MMFF94 (*OEMMFFParams*) or MMFF94s (*OEMMFF94sParams*) force field, or a different set of user defined parameters (*OEFFParams*) during construction.

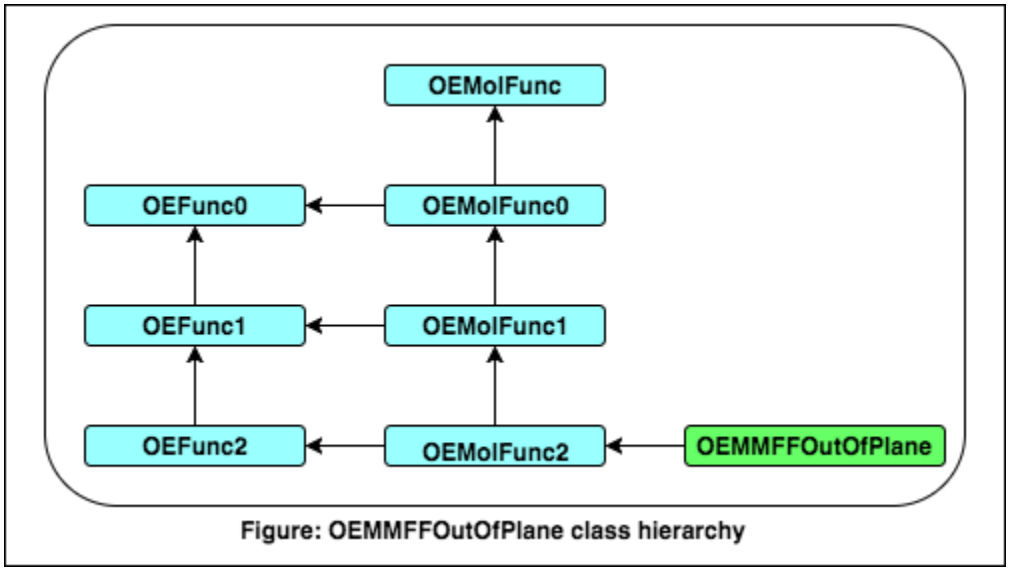


Figure: OEMMFFOutOfPlane class hierarchy

The following methods are publicly inherited from *OEFunc0*:

- *operator()*
- *GetFComponents*
- *NumVar*

The following methods are publicly inherited from *OEFunc1*:

- *operator()*

The following methods are publicly inherited from *OEFunc2*:

- *operator()*

The following methods are publicly inherited from *OEMolFunc*:

- *Clear*
- *ClearPredicates*
- *GetInteractions*
- *Setup*

The *OEMMFFOutOfPlane* class defines the following public methods:

Constructors

```
OEMMFFOutOfPlane(const OEMMFFOutOfPlane &)
OEMMFFOutOfPlane(const OEMolPotential::OEFFParams &)
```

Default and copy constructors.

Constructs an *OEMMFFOutOfPlane* instance using the specified set of parameters.

3.5.4 OEMMFFParams

> **Attention**
>
> This API is currently available in C++ and Python.

```
class OEMMFFParams : public OEMolPotential::OEFFParams
```

The *OEMMFFParams* class provides the ability to retrieve parameters used in the MMFF94 force field.

The following methods are publicly inherited from *OEFFParams*:

- *CreateCopy*
- *GetCoulombParams*
- *GetStretchParams*
- *GetAtomTypeIndex*
- *GetEquilibriumAngle*
- *GetTorsionParams*
- *GetAtomTypeName*
- *GetEquilibriumBondLength*
- *GetVdwParams*
- *GetBendParams*
- *GetOutOfPlaneParams*
- *GetVdwRadius*
- *GetContactDistance*
- *GetStretchBendParams*
- *PrepMol*
- *GetVdwParams*
- *GetStretchParams*
- *GetBendParams*
- *GetTorsionParams*
- *GetOutOfPlaneParams*
- *HasVdwParams*
- *HasStretchParams*

- *HasBendParams*
- *HasTorsionParams*
- *GetTitle*

The following classes derive from this class:

- *OEMMFF94sParams*

Constructors

```
OEMMFFParams()
```

Default and copy constructors.

3.5.5 OEMMFFStretchBend

> **Attention**
>
> This API is currently available in C++ and Python.

```
class OEMMFFStretchBend : public OEMolPotential::OEMolFunc2
```

The *OEMMFFStretchBend* class defines stretch-bend interaction function using the functional forms used in MMFF94 force field. The function can be initialized with parameters defined in MMFF94 (*OEMMFFParams*) or MMFF94s (*OEMMFF94sParams*) force field, or a different set of user defined parameters (*OEFFParams*) during construction.

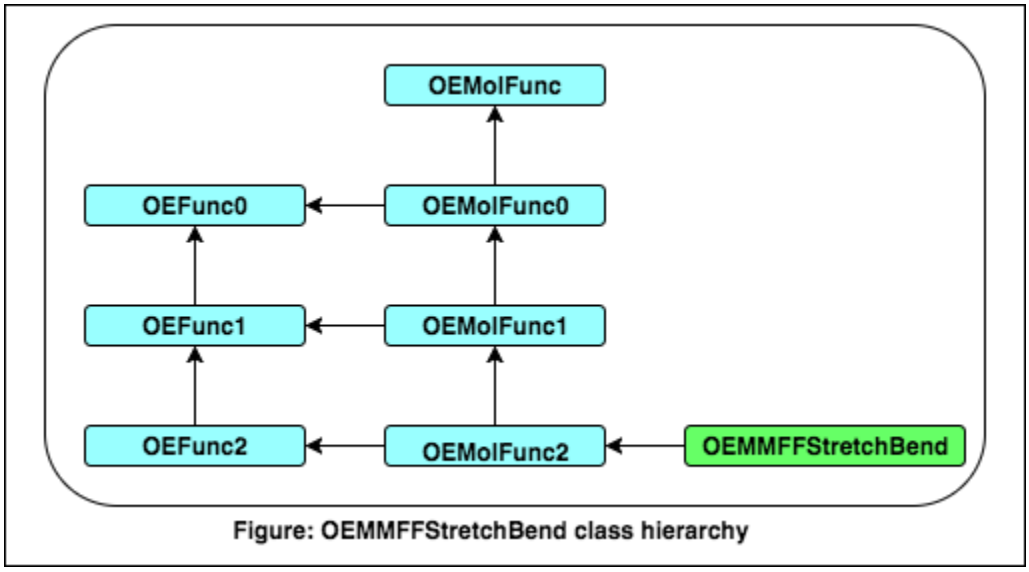


Figure: OEMMFFStretchBend class hierarchy

The following methods are publicly inherited from *OEFunc0*:

- *operator()*
- *GetFComponents*
- *NumVar*

The following methods are publicly inherited from *OEFunc1*:

- *operator()*

The following methods are publicly inherited from *OEFunc2*:

- *operator()*

The following methods are publicly inherited from *OEMolFunc*:

- *Clear*
- *ClearPredicates*
- *GetInteractions*
- *Setup*

The *OEMMFFStretchBend* class defines the following public methods:

- *Set*

Constructors

```
OEMMFFStretchBend(const OEMMFFStretchBend &)
OEMMFFStretchBend(const OEMolPotential::OEFFParams &)
```

Default and copy constructors.

Constructs an *OEMMFFStretchBend* instance using the specified set of parameters.

Set

```
bool Set(const OESystem::OEBinaryPredicate<OEChem::OEBondBase, OEChem::OEBondBase>&,
         unsigned int flag = OEFuncType::StretchBend)
```

This method defines the set of stretch-bending interactions that should not be included during the *Setup*. *Set* must be called before the *Setup* for it to be effective.

3.6 OEMolMMFF Functions

3.6.1 OEIsValidMMFFMolecule

> **Attention**
>
> This API is currently available in C++ and Python.

```
bool OEIsValidMMFFMolecule(const OEChem::OEMolBase& mol,
                           const OEMMFFParams& param = OEMMFF94sParams(),
                           const bool ignoreCharges = false)
```

This free function checks if MMFF parameters are available for the specified molecule. If the optional argument `param` is provided, check is performed against that specific *OEMMFFParams*, otherwise the *OEMMFF94sParams* is used by default. The second optional parameter `ignoreCharges` allows to ignore MMFF charge assigment while check is performed.

3.6.2 OEMMFFCalcVdw

> **Attention**
>
> This API is currently available in C++ and Python.

```
double OEMMFFCalcVdw(const OEChem::OEMolBase& mol,
                     const OEChem::OEAtomBase* patom,
                     const OEChem::OEAtomBase* latom,
                     bool coloumb = false, bool attract = false)
```

This free function calculates the VdW/Coulomb interaction energy between a pair of atoms passed to the function as the second and third arguments, belonging to a molecule `mol`. Coulomb term is added to the result only when the fourth argument to the function is set `true`. A regular (not modified) MMFF VdW potential is used when the value of the last argument is set to `true`.

3.7 OEMolSmirnoff Classes

3.7.1 OEParsley

> **Attention**
>
> This API is currently available in C++ and Python.

```
class OEParsley : public OESmirnoff
```

The *OEParsley* class facilitates creation of an instance of Parsley force field.

The following methods are publicly inherited from *OEFunc0*:

- *operator()*
- *GetFComponents*
- *NumVar*

The following methods are publicly inherited from *OEFunc1*:

- *operator()*

The following methods are publicly inherited from *OEFunc2*:

- *operator()*

The following methods are publicly inherited from *OEMolFunc*:

- *Clear*
- *ClearPredicates*
- *GetInteractions*
- *Setup*

The following methods are publicly inherited from *OEForceField*:

- *GetTitle* and *SetTitle*

- *HasCoulomb*
- *HasInternalCharges*
- *PrepMol*
- *Set*
- *SetNonBondCutoff*

The following methods are publicly inherited from *OEGenericFF2*:

- *AddMolFunc*
- *GetComponents*
- *RemoveMolFunc*

Constructors

```
OEParsley(const OEMolPotential::OENonBondIntcsOptions& opts =␣
↪OEMolPotential::OENonBondIntcsOptions())
OEParsley(const OEParsley&)
```

Default and copy constructors.

3.7.2 OEParsleyParams

> **Attention**
>
> This API is currently available in C++ and Python.

```
class OEParsleyParams : public OESmirnoffParams
```

The *OEParsleyParams* class provides the ability to retrieve parameters necessary to perform energy and gradient calculations using the *OEParsley* force field.

The following methods are publicly inherited from *OEFFParams*:

- *CreateCopy*
- *GetCoulombParams*
- *GetStretchParams*
- *GetAtomTypeIndex*
- *GetEquilibriumAngle*
- *GetTorsionParams*
- *GetAtomTypeName*
- *GetEquilibriumBondLength*
- *GetVdwParams*
- *GetBendParams*
- *GetOutOfPlaneParams*
- *GetVdwRadius*

- *GetContactDistance*
- *GetStretchBendParams*
- *PrepMol*
- *GetVdwParams*
- *GetStretchParams*
- *GetBendParams*
- *GetTorsionParams*
- *GetOutOfPlaneParams*
- *HasVdwParams*
- *HasStretchParams*
- *HasBendParams*
- *HasTorsionParams*
- *GetTitle*

Constructors

```
OEParsleyParams()
OEParsleyParams(const OEParsleyParams&)
```

Default and copy constructors.

3.7.3 OESage

> **Attention**
>
> This API is currently available in C++ and Python.

```
class OESage : public OESmirnoff
```

The *OESage* class facilitates creation of an instance of Sage force field.

The following methods are publicly inherited from *OEFunc0*:

- *operator()*
- *GetFComponents*
- *NumVar*

The following methods are publicly inherited from *OEFunc1*:

- *operator()*

The following methods are publicly inherited from *OEFunc2*:

- *operator()*

The following methods are publicly inherited from *OEMolFunc*:

- *Clear*
- *ClearPredicates*

- *GetInteractions*
- *Setup*

The following methods are publicly inherited from *OEForceField*:

- *GetTitle* and *SetTitle*
- *HasCoulomb*
- *HasInternalCharges*
- *PrepMol*
- *Set*
- *SetNonBondCutoff*

The following methods are publicly inherited from *OEGenericFF2*:

- *AddMolFunc*
- *GetComponents*
- *RemoveMolFunc*

Constructors

```
OESage(const OEMolPotential::OENonBondIntcsOptions& opts =␣
↪OEMolPotential::OENonBondIntcsOptions())
OESage(const OESage&)
```

Default and copy constructors.

3.7.4 OESageParams

> **Attention**
>
> This API is currently available in C++ and Python.

```
class OESageParams : public OESmirnoffParams
```

The *OESageParams* class provides the ability to retrieve parameters necessary to perform energy and gradient calculations using the *OESage* force field.

The following methods are publicly inherited from *OEFFParams*:

- *CreateCopy*
- *GetCoulombParams*
- *GetStretchParams*
- *GetAtomTypeIndex*
- *GetEquilibriumAngle*
- *GetTorsionParams*
- *GetAtomTypeName*
- *GetEquilibriumBondLength*

- *GetVdwParams*
- *GetBendParams*
- *GetOutOfPlaneParams*
- *GetVdwRadius*
- *GetContactDistance*
- *GetStretchBendParams*
- *PrepMol*
- *GetVdwParams*
- *GetStretchParams*
- *GetBendParams*
- *GetTorsionParams*
- *GetOutOfPlaneParams*
- *HasVdwParams*
- *HasStretchParams*
- *HasBendParams*
- *HasTorsionParams*
- *GetTitle*

Constructors

```
OESageParams()
OESageParams(const OESageParams&)
```

Default and copy constructors.

3.7.5 OESmirnoff

> **Attention**
>
> This API is currently available in C++ and Python.

```
class OESmirnoff : public OEMolPotential::OEGenericFF2
```

The *OESmirnoff* class facilitates creation of an instance of SMIRNOFF force field (see *SMIRNOFF*). In addition, a force field instance can be extended by adding externally defined potential functions.

The following methods are publicly inherited from *OEFunc0*:

- *operator()*
- *GetFComponents*
- *NumVar*

The following methods are publicly inherited from *OEFunc1*:

- *operator()*

The following methods are publicly inherited from *OEFunc2*:

- *operator()*

The following methods are publicly inherited from *OEMolFunc*:

- *Clear*
- *ClearPredicates*
- *GetInteractions*
- *Setup*

The following methods are publicly inherited from *OEForceField*:

- *GetTitle* and *SetTitle*
- *HasCoulomb*
- *HasInternalCharges*
- *PrepMol*
- *Set*
- *SetNonBondCutoff*

The following methods are publicly inherited from *OEGenericFF2*:

- *AddMolFunc*
- *GetComponents*
- *RemoveMolFunc*

Constructors

```
OESmirnoff(const OESmirnoffParams& params,
           const OEMolPotential::OENonBondIntcsOptions& opts =␣
↪OEMolPotential::OENonBondIntcsOptions())
OESmirnoff(const OESmirnoff&)
```

Default and copy constructors.

3.7.6 OESmirnoffParams

> **Attention**
>
> This API is currently available in C++ and Python.

```
class OESmirnoffParams : public OEMolPotential::OEFFParams
```

The OESmirnoffParams class provides the ability to retrieve parameters necessary to perform energy and gradient calculations using the SMIRNOFF force field (see *SMIRNOFF*). By default OESmirnoffParams loads the `Smirnoff99Frost` force field.

> **Note**
>
> The inherited methods, *GetAtomTypeIndex*, *GetAtomTypeName* and *GetStretchBendParams*, have no meaning in the SMIRNOFF force field and SHOULD NOT be used. When these methods are called, they return `0`, `none`, and `false`, respectively.

The following methods are publicly inherited from *OEFFParams*:

- *CreateCopy*
- *GetCoulombParams*
- *GetStretchParams*
- *GetAtomTypeIndex*
- *GetEquilibriumAngle*
- *GetTorsionParams*
- *GetAtomTypeName*
- *GetEquilibriumBondLength*
- *GetVdwParams*
- *GetBendParams*
- *GetOutOfPlaneParams*
- *GetVdwRadius*
- *GetContactDistance*
- *GetStretchBendParams*
- *PrepMol*
- *GetVdwParams*
- *GetStretchParams*
- *GetBendParams*
- *GetTorsionParams*
- *GetOutOfPlaneParams*
- *HasVdwParams*
- *HasStretchParams*
- *HasBendParams*
- *HasTorsionParams*
- *GetTitle*

The *OESmirnoffParams* class contains the following methods:

- *AddBendData*
- *AddImproperData*
- *AddStretchData*
- *AddTorsionData*

- *AddVdWData*
- *GetBendData*
- *GetImproperData*
- *GetStretchData*
- *GetTorsionData*
- *GetVdWData*
- *Load*
- *SetBendData*
- *SetImproperData*
- *SetStretchData*
- *SetTorsionData*
- *SetVdWData*

Constructors

```
OESmirnoffParams()
OESmirnoffParams(const OESmirnoffParams&)
```

Default and copy constructors.

Assignment operator.

3.7.7 AddBendData

```
bool AddBendData(const OESmirnoffBendData& data)
```

Adds new bending parameters based on provided *OESmirnoffBendData*.

3.7.8 AddImproperData

```
bool AddImproperData(const OESmirnoffImproperData& data)
```

Adds new improper torsion parameters based on provided *OESmirnoffImproperData*.

3.7.9 AddStretchData

```
bool AddStretchData(const OESmirnoffStretchData& data)
```

Adds new bond stretching parameters based on provided *OESmirnoffStretchData*.

3.7.10 AddTorsionData

```
bool AddTorsionData(const OESmirnoffTorsionData& data)
```

Adds new torsion parameters based on provided *OESmirnoffTorsionData*.

3.7.11 AddVdWData

```
bool AddVdWData(const OESmirnoffVdWData& data)
```

Adds new Van der Waals parameters based on provided *OESmirnoffVdWData*.

3.7.12 GetBendData

```
const std::vector<OESmirnoffBendData>& GetBendData() const
bool GetBendData(OESmirnoffBendData& data,
                 const OEChem::OEAtomBase*,
                 const OEChem::OEAtomBase*,
                 const OEChem::OEAtomBase*) const
```

The first overload provides all of the *OESmirnoffBendData* corresponding to the bending parameters in the forcefield. The second overload can be used to access parameters for the angle specified by the three atoms.

3.7.13 GetImproperData

```
const std::vector<OESmirnoffImproperData>& GetImproperData() const
bool GetImproperData(OESmirnoffImproperData& data,
                     const OEChem::OEAtomBase*,
                     const OEChem::OEAtomBase*,
                     const OEChem::OEAtomBase*,
                     const OEChem::OEAtomBase*) const
```

The first overload provides all of the *OESmirnoffImproperData* corresponding to the improper torsion parameters in the forcefield. The second overload can be used to access parameters for the plane specified by the four atoms. The first atom must be the central atom of an improper torsion.

3.7.14 GetStretchData

```
const std::vector<OESmirnoffStretchData>& GetStretchData() const
bool GetStretchData(OESmirnoffStretchData& data,
                    const OEChem::OEBondBase*) const
```

The first overload provides all of the *OESmirnoffStretchData* corresponding to the bond stretching parameters in the forcefield. The second overload can be used to access parameters for the specified bond.

3.7.15 GetTorsionData

```
const std::vector<OESmirnoffTorsionData>& GetTorsionData() const
bool GetTorsionData(OESmirnoffTorsionData& data,
                    const OEChem::OEAtomBase*,
                    const OEChem::OEAtomBase*,
                    const OEChem::OEAtomBase*,
                    const OEChem::OEAtomBase*) const
```

The first overload provides all of the *OESmirnoffTorsionData* corresponding to the torsion parameters in the forcefield. The second overload can be used to access parameters for the torsion specified by the four atoms.

3.7.16 GetVdWData

```
const std::vector<OESmirnoffVdWData>& GetVdWData() const
bool GetVdWData(OESmirnoffVdWData& data, const OEChem::OEAtomBase*) const
```

The first overload provides all of the *OESmirnoffVdWData* corresponding to the VdW parameters in the forcefield. The second overload can be used to access parameters for the specified atom.

3.7.17 Load

```
bool Load(const unsigned int fftype)
bool Load(const std::string& xml_file_name)
bool Load(const OEPlatform::oeisstream& ifs)
```

Methods which load force field parameters. The first method is used to load available build-in parameter set, defined in the *OESmirnoffType* namespace. The second method takes the name of the parameter file in XML format. This file might contain user custom parameters. The third method takes a stream object `oeisstream` which contains Smirnoff parameters. It is a convenience method which might be used in user code when the parameters XML file is converted a stream.

3.7.18 SetBendData

```
bool SetBendData(const OESmirnoffBendData& data)
```

Sets the *OESmirnoffBendData* replacing existing parameters with same ID and Smarts.

3.7.19 SetImproperData

```
bool SetImproperData(const OESmirnoffImproperData& data)
```

Sets the *OESmirnoffImproperData* replacing existing parameters with same ID and Smarts.

3.7.20 SetStretchData

```
bool SetStretchData(const OESmirnoffStretchData& data)
```

Sets the *OESmirnoffStretchData* replacing existing parameters with same ID and Smarts.

3.7.21 SetTorsionData

```
bool SetTorsionData(const OESmirnoffTorsionData& data)
```

Sets the *OESmirnoffTorsionData* replacing existing parameters with same ID and Smarts.

3.7.22 SetVdWData

```
bool SetVdWData(const OESmirnoffVdWData& data)
```

Sets the *OESmirnoffVdWData* replacing existing parameters with same ID and Smarts.

3.7.23 OESmirnoffData

Attention

This API is currently available in C++ and Python.

```
class OESmirnoffData
```

The OESmirnoffData class provides an interface to the parameters in the SMIRNOFF force field (see *SMIRNOFF*).

See also

- *OESmirnoffParams* class

The class defines the following public methods:

- *GetID*
- *GetSmarts*

GetID

```
const std::string& GetID() const
```

Returns the unique id for the parameter

GetSmarts

```
const std::string& GetSmarts() const
```

Returns the SMARTS pattern for the parameter

3.7.24 OESmirnoffStretchData

Attention

This API is currently available in C++ and Python.

```
class OESmirnoffStretchData : public OEMolSmirnoff::OESmirnoffData
```

The OESmirnoffStretchData class provides the ability to add new parameters and modify existing bond stretching parameters in the SMIRNOFF force field that has the functional form: $V_b = k_b(r - r_0)^2$ (see *SMIRNOFF*).

See also

- *OESmirnoffData* class

The following methods are publicly inherited from *OESmirnoffData* class:

- *GetID*
- *GetSmarts*

The class defines the following public methods:

- *GetEquilibriumBondLength*
- *GetEneConst*
- *SetParams*

Constructors

```
OESmirnoffStretchData()
OESmirnoffStretchData(const std::string& id,
                      const std::string& smarts)
OESmirnoffStretchData(const OESmirnoffStretchData&)
```

GetEquilibriumBondLength

```
double GetEquilibriumBondLength() const
```

Returns the equilibrium bond length (r_0) in Angstrom associated with the parameter

GetEneConst

```
double GetEneConst() const
```

Returns the energy constant (k_b) $kcal/mol/Angstrom^2$ associated with the parameter

SetParams

```
bool SetParams(const double eneConst, const double bondLength)
```

Sets equilibrium bond length and energy constant for the the parameter

3.7.25 OESmirnoffBendData

Attention

This API is currently available in C++ and Python.

```
class OESmirnoffBendData : public OEMolSmirnoff::OESmirnoffData
```

The OESmirnoffBendData class provides the ability to add new parameters and modify existing angle bending parameters in the SMIRNOFF force field that has the functional form: $V_a = k_a(\alpha - \alpha_0)^2$ (see *SMIRNOFF*).

See also

- *OESmirnoffData* class

The following methods are publicly inherited from *OESmirnoffData* class:

- *GetID*
- *GetSmarts*

The class defines the following public methods:

- *GetEquilibriumAngle*
- *GetEneConst*
- *SetParams*

Constructors

```
OESmirnoffBendData()
OESmirnoffBendData(const std::string& id, const std::string& smarts)
OESmirnoffBendData(const OESmirnoffBendData&)
```

GetEquilibriumAngle

```
double GetEquilibriumAngle() const
```

Returns the equilibrium angle (α_0) in degrees associated with the parameter

GetEneConst

```
double GetEneConst() const
```

Returns the energy constant (k_a) kcal/mol/radian associated with the parameter

SetParams

```
bool SetParams(const double eneConst, const double angle)
```

Sets equilibrium angle (α_0) in degrees and energy constant (k_a) kcal/mol/radian for the the parameter

3.7.26 OESmirnoffTorsionData

Attention

This API is currently available in C++ and Python.

```
class OESmirnoffTorsionData : public OEMolSmirnoff::OESmirnoffData
```

The OESmirnoffTorsionData class provides the ability to add new parameters and modify existing proper torsion parameters in the SMIRNOFF force field that has the functional form: $V_t = \sum_n V_n/d_n(1 + \cos(p_n\theta - \gamma_n))$ (see *SMIRNOFF*).

See also

- *OESmirnoffData* class

The following methods are publicly inherited from *OESmirnoffData* class:

- *GetID*
- *GetSmarts*

The class defines the following public methods:

- *NumTerms*
- *GetDegeneracy*
- *GetPeriodicity*
- *GetEneConst*
- *GetPhaseAngle*
- *AddData*
- *ClearData*

Constructors

```
OESmirnoffTorsionData()
OESmirnoffTorsionData(const std::string& id,
                      const std::string& smarts)
OESmirnoffTorsionData(const OESmirnoffTorsionData&)
```

NumTerms

```
unsigned NumTerms() const
```

Returns the total number of terms (n)

GetDegeneracy

```
unsigned GetDegeneracy(const unsigned index) const
```

Returns the degeneracy (d_n) of the OESmirnoffTorsionData term corresponding to the index

GetPeriodicity

```
unsigned GetPeriodicity(const unsigned index) const
```

Returns the periodicity of the cosine term (p_n) of the OESmirnoffTorsionData term corresponding to the index

GetEneConst

```
double GetEneConst(const unsigned index) const
```

Returns the energy constant (V_n) in kcal/mol multiplied to the cosine term of the OESmirnoffTorsionData term corresponding to the index

GetPhaseAngle

```
double GetPhaseAngle(const unsigned index) const
```

Returns the phase (γ_n) in degrees of the cosine term of the OESmirnoffTorsionData term corresponding to the index

AddData

```
bool AddData(const unsigned degeneracy, const unsigned periodicity,
             const double eneConst, const double phaseAngle)
```

Used to add a OESmirnoffTorsionData term to a set of OESmirnoffParams

ClearData

```
bool ClearData()
```

Clears the degeneracy, periodicity, energy constant and phase from a OESmirnoffTorsionData. The ID and Smarts associated with the OESmirnoffTorsionData are not cleared.

3.7.27 OESmirnoffImproperData

Attention

This API is currently available in C++ and Python.

```
class OESmirnoffImproperData : public OEMolSmirnoff::OESmirnoffData
```

The OESmirnoffImproperData class provides the ability to add new parameters and modify existing improper torsion (out-of-plane bending) parameters in the SMIRNOFF force field that has the functional form: $V_o = V(1+\cos(p\theta - \gamma))$ (see *SMIRNOFF*)

See also

- *OESmirnoffData* class

The following methods are publicly inherited from *OESmirnoffData* class:

- *GetID*
- *GetSmarts*

The class defines the following public methods:

- *GetEneConst*
- *GetPhaseAngle*
- *GetPeriodicity*
- *SetParams*

Constructors

```
OESmirnoffImproperData()
OESmirnoffImproperData(const std::string& id,
                       const std::string& smarts)
OESmirnoffImproperData(const OESmirnoffImproperData&)
```

GetEneConst

```
double GetEneConst() const
```

Returns the energy constant (V_n) in kcal/mol for the term

GetPhaseAngle

```
double GetPhaseAngle() const
```

Returns the phase (γ_n) in degrees for the term

GetPeriodicity

```
unsigned GetPeriodicity() const
```

Returns the periodicity (p_n) of the term

SetParams

```
bool SetParams(const double eneConst, const double phase,
               const unsigned periodicity)
```

Sets the energy constant (V_n) , phase (γ_n) and periodicity (p_n) of the improper torsion parameter.

3.7.28 OESmirnoffVdWData

Attention

This API is currently available in C++ and Python.

```
class OESmirnoffVdWData : public OEMolSmirnoff::OESmirnoffData
```

The OESmirnoffVdWData class provides the ability to add new parameters and modify existing proper torsion parameters in the SMIRNOFF force field that has the functional form: $V_v = \epsilon_{ij} \left[\left(\frac{R_{ij}}{r_{ij}} \right)^{12} - 2 \left(\frac{R_{ij}}{r_{ij}} \right)^{6} \right]$ (see *SMIRNOFF*).

See also

- *OESmirnoffData* class

The following methods are publicly inherited from *OESmirnoffData* class:

- *GetID*
- *GetSmarts*

The class defines the following public methods:

- *GetEpsilon*
- *GetRadius*
- *SetParams*

Constructors

```
OESmirnoffVdWData()
OESmirnoffVdWData(const std::string& id, const std::string& smarts)
OESmirnoffVdWData(const OESmirnoffVdWData&)
```

GetEpsilon

```
double GetEpsilon() const
```

Returns the epsilon value (ϵ_{ij}) in kcal/mol for VdW term

GetRadius

```
double GetRadius() const
```

Returns the radius in (R_{ij}) Angstrom associated with VdW term

SetParams

```
bool SetParams(const double eplison, const double radius)
```

Sets the epsilon and radius of the VdW interaction term

3.8 OEMolSmirnoff Constants

3.8.1 OESmirnoffType

This namespace contains constants that are used in *OESmirnoffParams.Load* to define a specific build-in set of paramaters which might be used.

SMIRNOFF99FROSST

SMIRNOFF99FROSST set of parameters

PARSLEY_OPENFF

Parsley OpenFF1.3.1 set of parameters

SAGE_OPENFF

Latest version of the Sage set of parameters

3.9 OESheff Classes

3.9.1 OESheffield

Attention

This API is currently available in C++ and Python.

```
class OESheffield : public OEMolPotential::OEMolFunc2
```

This class represents *OESheffield*.

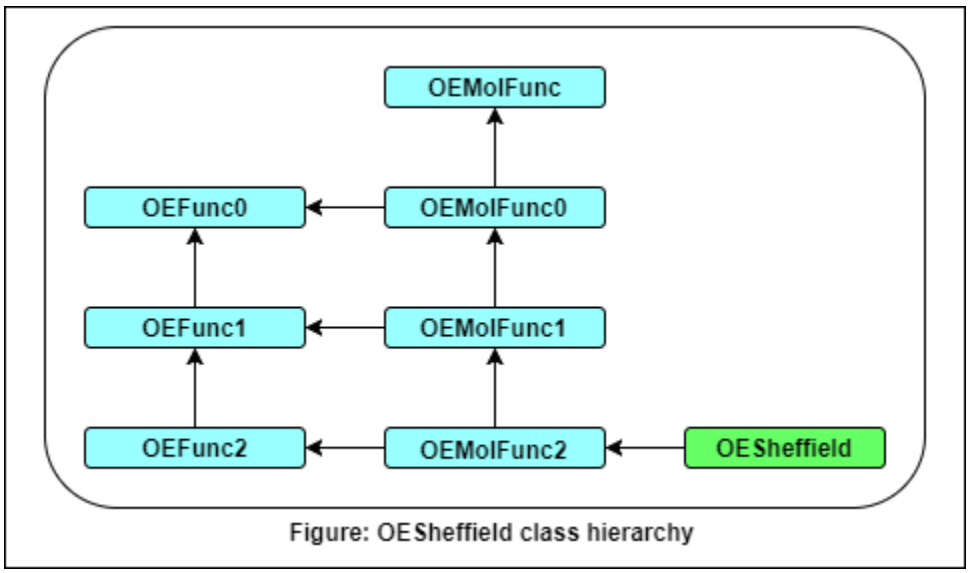


Figure: OESheffield class hierarchy

The following methods are publicly inherited from *OEFunc0*:

- *operator()*
- *GetFComponents*
- *NumVar*

The following methods are publicly inherited from *OEFunc1*:

- *operator()*

The following methods are publicly inherited from *OEFunc2*:

- *operator()*

The following methods are publicly inherited from *OEMolFunc*:

- *Clear*
- *ClearPredicates*
- *GetInteractions*
- *Setup*

The *OESheffield* class defines the following public methods:

- *GetDielectricDerivative*
- *Set*
- *SetCharges*

- *SetRadii*

Constructors

```
OESheffield(double aa = 1.553149, double bb = 0.735694)
OESheffield(const OESheffieldOptions&)
OESheffield(const OESheffield&)
```

Default and copy constructors.

The first constructor uses default options and the second uses the user specific options. The default Sheffield model parameters are set according to Grant *et. al.* ([Grant-2007]).

GetDielectricDerivative

```
double GetDielectricDerivative(const double* x, double solv_diel) const
```

Returns the derivative of the solvation energy with respect to the solvent dielectric constant.

Set

```
bool Set(const OESystem::OEUnaryPredicate<OEChem::OEAtomBase>&,
         unsigned int = OEFuncType::Sheffield)
bool Set(const OESystem::OEBinaryPredicate<OEChem::OEAtomBase, OEChem::OEAtomBase>&,
         unsigned int = OEFuncType::Sheffield)
```

Sets the unary and binary predicates respectively which might be used to modify the results of the calculations. Unary predicate might be used to selectively exclude a subset of atoms from the calculation of molecule "self" energy, while the binary predicate can include only selected pairs of atoms in the calculation of solvation energy. By default when no predicates are set, atomic pairs are included in the solvation energy of a molecule.

SetCharges

```
void SetCharges(float*)
```

Atomic charges are taken from the provided external array.

SetRadii

```
void SetRadii(float*)
```

Atomic radii values are taken from the provided external array.

3.9.2 OESheffieldOptions

> **Attention**
>
> This API is currently available in C++ and Python.

```
class OESheffieldOptions : public OESystem::OEOptions
```

This class orovides interface to setup options required for *OESheffield* calculations.

The *OESheffieldOptions* class defines the following public methods:

- *GetA*
- *GetB*
- *GetC*
- *GetSoluteDielectricConstant*
- *GetSolventDielectricConstant*
- *SetA*
- *SetB*
- *SetC*
- *SetSoluteDielectricConstant*
- *SetSolventDielectricConstant*

Constructors

```
OESheffieldOptions(const unsigned type = OESheffieldParamType::SmallMol);
OESheffieldOptions(const OESheffieldOptions&)
```

Default and copy constructors.

GetA

```
double GetA() const
```

Gets the first Sheffield parameter value.

GetB

```
double GetB() const
```

Gets the second Sheffield parameter value.

GetC

```
double GetC() const
```

Gets the third Sheffield parameter value.

GetSoluteDielectricConstant

```
double GetSoluteDielectricConstant() const
```

Gets the dielectric constant of the solute.

GetSolventDielectricConstant

```
double GetSolventDielectricConstant() const
```

Gets the dielectric constant of the solvent.

SetA

```
bool SetA(const double a)
```

Optionally changes the first Sheffield parameter to the passed value a.

SetB

```
bool SetB(const double b)
```

Optionally changes the second Sheffield parameter to the passed value b.

SetC

```
bool SetC(const double c)
```

Optionally changes the third Sheffield parameter to the passed value c.

SetSoluteDielectricConstant

```
bool SetSoluteDielectricConstant(const double epsilon)
```

Sets the dielectric constant for the solute.

SetSolventDielectricConstant

```
bool SetSolventDielectricConstant(const double eps)
```

Sets the dielectric constant for the solvent.

3.10 OESheff Constants

3.10.1 OESheffieldParamType

This namespace contains constants that defines which set of Sheffield parameters are to be loaded in construction of *OESheffieldOptions*.

SmallMol

Classical Sheffield parameters suitable for small-molecules

Protein

Modified Sheffield parameters suitable for proteins and complexes

3.11 OEFF Classes

3.11.1 OEFF14SBParsley

> **Attention**
>
> This API is currently available in C++ and Python.

```
class OEFF14SBParsley : public OEFF14SBSmirnoff
```

The *OEFF14SBParsley* is a class for protein systems which might contain cofactors. It assumes that the iteraction are described with **OEFF14SBParsleyParams**.

The following methods are publicly inherited from *OEFunc0*:

- *operator()*
- *GetFComponents*
- *NumVar*

The following methods are publicly inherited from *OEFunc1*:

- *operator()*

The following methods are publicly inherited from *OEFunc2*:

- *operator()*

The following methods are publicly inherited from *OEMolFunc*:

- *Clear*
- *ClearPredicates*
- *GetInteractions*
- *Setup*

The following methods are publicly inherited from *OEForceField*:

- *Set*

The following methods are publicly inherited from *OEGenericFF2*:

- *AddMolFunc*
- *GetComponents*
- *RemoveMolFunc*

The following methods are publicly inherited from *OEFF14SBSmirnoff*

- *PrepMol*

Constructors

```
OEFF14SBParsley(const OEMolPotential::OENonBondIntcsOptions& opts =␣
↪OEMolPotential::OENonBondIntcsOptions())
OEFF14SBParsley(const OEFF14SBParsley&)
```

Default and copy constructors.

3.11.2 OEFF14SBParsleyComplex

> **Attention**
>
> This API is currently available in C++ and Python.

```
class OEFF14SBParsleyComplex : public OEComplexFF
```

The *OEFF14SBParsleyComplex* represents a force field for a protein-ligand complex system, where the interactions are described with **OEFF14SBParsleyParams**.

The following methods are publicly inherited from *OEFunc0*:

- *operator()*
- *GetFComponents*
- *NumVar*

The following methods are publicly inherited from *OEFunc1*:

- *operator()*

The following methods are publicly inherited from *OEMolFunc*:

- *Clear*
- *ClearPredicates*
- *GetInteractions*
- *Setup*

The following methods are publicly inherited from *OEComplexFF*:

- *AdaptHostCoords*
- *GetVar*
- *Set*
- *SetHostFlex*
- *SetNonBondCutoff*
- *SetupHost*

Constructors

```
OEFF14SBParsleyComplex(const OEMolPotential::OENonBondIntcsOptions& opts =␣
↪OEMolPotential::OENonBondIntcsOptions())
OEFF14SBParsleyComplex(const OEFF14SBParsleyComplex&)
```

Default and copy constructors.

3.11.3 OEFF14SBParsleyParams

Attention

This API is currently available in C++ and Python.

```
class OEFF14SBParsleyParams : public public OEFF14SBSmirnoffParams
```

The *OEFF14SBParsleyParams* class provides the ability to retrieve parameters necessary to perform energy and gradient calculations using the OEFF14SBParsley force field.

Note

The inherited methods, *GetAtomTypeIndex*, *GetAtomTypeName* and *GetStretchBendParams*, have no meaning in the Parsley force field and SHOULD NOT be used.

The following methods are publicly inherited from *OEFFParams*:

- *CreateCopy*
- *GetCoulombParams*
- *GetStretchParams*
- *GetAtomTypeIndex*
- *GetEquilibriumAngle*
- *GetTorsionParams*
- *GetAtomTypeName*
- *GetEquilibriumBondLength*
- *GetVdwParams*
- *GetBendParams*
- *GetOutOfPlaneParams*
- *GetVdwRadius*
- *GetContactDistance*
- *GetStretchBendParams*
- *PrepMol*
- *GetVdwParams*
- *GetStretchParams*
- *GetBendParams*
- *GetTorsionParams*
- *GetOutOfPlaneParams*
- *HasVdwParams*
- *HasStretchParams*
- *HasBendParams*
- *HasTorsionParams*

- *GetTitle*

The following methods are publicly inherited from *OEFF14SBSmirnoffParams*:

- *Load*

Constructors

```
OEFF14SBParsleyParams()
OEFF14SBParsleyParams(const OEFF14SBParsleyParams&)
```

Default and copy constructors.

3.11.4 OEFF14SBSage

> **Attention**
>
> This API is currently available in C++ and Python.

```
class OEFF14SBSage : public OEFF14SBSmirnoff
```

The *OEFF14SBSage* is a class for protein systems which might contain cofactors. It assumes that the iteraction are described with **OEFF14SBSageParams**.

The following methods are publicly inherited from *OEFunc0*:

- *operator()*
- *GetFComponents*
- *NumVar*

The following methods are publicly inherited from *OEFunc1*:

- *operator()*

The following methods are publicly inherited from *OEFunc2*:

- *operator()*

The following methods are publicly inherited from *OEMolFunc*:

- *Clear*
- *ClearPredicates*
- *GetInteractions*
- *Setup*

The following methods are publicly inherited from *OEForceField*:

- *Set*

The following methods are publicly inherited from *OEGenericFF2*:

- *AddMolFunc*
- *GetComponents*
- *RemoveMolFunc*

The following methods are publicly inherited from *OEFF14SBSmirnoff*

- *PrepMol*

Constructors

```
OEFF14SBSage(const OEMolPotential::OENonBondIntcsOptions& opts =␣
↪OEMolPotential::OENonBondIntcsOptions())
OEFF14SBSage(const OEFF14SBSage&)
```

Default and copy constructors.

3.11.5 OEFF14SBSageComplex

> **Attention**
>
> This API is currently available in C++ and Python.

```
class OEFF14SBSageComplex : public OEComplexFF
```

The *OEFF14SBSageComplex* represents a force field for a protein-ligand complex system, where the interactions are described with **OEFF14SBSageParams**.

The following methods are publicly inherited from *OEFunc0*:

- *operator()*
- *GetFComponents*
- *NumVar*

The following methods are publicly inherited from *OEFunc1*:

- *operator()*

The following methods are publicly inherited from *OEMolFunc*:

- *Clear*
- *ClearPredicates*
- *GetInteractions*
- *Setup*

The following methods are publicly inherited from *OEComplexFF*:

- *AdaptHostCoords*
- *GetVar*
- *Set*
- *SetHostFlex*
- *SetNonBondCutoff*
- *SetupHost*

Constructors

```
OEFF14SBSageComplex(const OEMolPotential::OENonBondIntcsOptions& opts =␣
↪OEMolPotential::OENonBondIntcsOptions())
OEFF14SBSageComplex(const OEFF14SBSageComplex&)
```

Default and copy constructors.

3.11.6 OEFF14SBSageParams

> **Attention**
>
> This API is currently available in C++ and Python.

```
class OEFF14SBSageParams : public public OEFF14SBSmirnoffParams
```

The *OEFF14SBSageParams* class provides the ability to retrieve parameters necessary to perform energy and gradient calculations using the OEFF14SBSage force field.

> **Note**
>
> The inherited methods, *GetAtomTypeIndex*, *GetAtomTypeName* and *GetStretchBendParams*, have no meaning in the Sage force field and SHOULD NOT be used.

The following methods are publicly inherited from *OEFFParams*:

- *CreateCopy*
- *GetCoulombParams*
- *GetStretchParams*
- *GetAtomTypeIndex*
- *GetEquilibriumAngle*
- *GetTorsionParams*
- *GetAtomTypeName*
- *GetEquilibriumBondLength*
- *GetVdwParams*
- *GetBendParams*
- *GetOutOfPlaneParams*
- *GetVdwRadius*
- *GetContactDistance*
- *GetStretchBendParams*
- *PrepMol*
- *GetVdwParams*
- *GetStretchParams*
- *GetBendParams*

- *GetTorsionParams*
- *GetOutOfPlaneParams*
- *HasVdwParams*
- *HasStretchParams*
- *HasBendParams*
- *HasTorsionParams*
- *GetTitle*

The following methods are publicly inherited from *OEFF14SBSmirnoffParams*:

- *Load*

Constructors

```
OEFF14SBSageParams()
OEFF14SBSageParams(const OEFF14SBSageParams&)
```

Default and copy constructors.

3.11.7 OEFF14SBSmirnoff

> **Attention**
>
> This API is currently available in C++ and Python.

```
class OEFF14SBSmirnoff : public OEGenericFF2
```

The *OEFF14SBSmirnoff* is a generic class for protein systems which might contain cofactors. It assumes that the intetractions are descrined with **OEFF14SBSmirnoffParams**.

The following methods are publicly inherited from *OEFunc0*:

- *operator()*
- *GetFComponents*
- *NumVar*

The following methods are publicly inherited from *OEFunc1*:

- *operator()*

The following methods are publicly inherited from *OEFunc2*:

- *operator()*

The following methods are publicly inherited from *OEMolFunc*:

- *Clear*
- *ClearPredicates*
- *GetInteractions*
- *Setup*

The following methods are publicly inherited from *OEForceField*:

- *Set*

The following methods are publicly inherited from *OEGenericFF2*:

- *AddMolFunc*
- *GetComponents*
- *RemoveMolFunc*

The following classes derive from this class:

- *OEFF14SBParsley*
- *OEFF14SBSage*

Constructors

```
OEFF14SBSmirnoff(const OEFF14SBSmirnoffParams& params = OEFF14SBSmirnoffParams(),
                 const OEMolPotential::OENonBondIntcsOptions& opts =␣
↪OEMolPotential::OENonBondIntcsOptions())
OEFF14SBSmirnoff(const OEFF14SBSmirnoff&)
```

Default and copy constructors.

PrepMol

```
bool PrepMol(OEChem::OEMolBase& mol, bool sweep = true,
             bool warnOK = true) const
```

Prepares molecule to be handled with proper parameters

3.11.8 OEFF14SBSmirnoffComplex

> **Attention**
>
> This API is currently available in C++ and Python.

```
class OEFF14SBSmirnoffComplex : public OEComplexFF
```

The *OEFF14SBSmirnoffComplex* represents a force field for a protein-ligand complex system, where the interactions are described with **OEFF14SBSmirnoffParams**.

The following methods are publicly inherited from *OEFunc0*:

- *operator()*
- *GetFComponents*
- *NumVar*

The following methods are publicly inherited from *OEFunc1*:

- *operator()*

The following methods are publicly inherited from *OEMolFunc*:

- *Clear*
- *ClearPredicates*

- *GetInteractions*
- *Setup*

The following methods are publicly inherited from *OEComplexFF*:

- *AdaptHostCoords*
- *GetVar*
- *Set*
- *SetHostFlex*
- *SetNonBondCutoff*
- *SetupHost*

Constructors

```
OEFF14SBSmirnoffComplex(const OEFF14SBSmirnoffParams& params = OEFF14SBSmirnoffParams(),
                        const OEMolPotential::OENonBondIntcsOptions& opts =␣
↪OEMolPotential::OENonBondIntcsOptions())
OEFF14SBSmirnoffComplex(const OEFF14SBSmirnoffComplex&)
```

Default and copy constructors.

3.11.9 OEFF14SBSmirnoffParams

Attention

This API is currently available in C++ and Python.

```
class OEFF14SBSmirnoffParams : public OEMolPotential::OEFFParams
```

The *OEFF14SBSmirnoffParams* class provides the ability to retrieve parameters necessary to perform energy and gradient calculations using OEFF14SBSmirnoff force field. By default parameters for the `Smirnoff99Frost` force field are used to handle nonprotein part of the molecular system.

Note

The inherited methods, *GetAtomTypeIndex*, *GetAtomTypeName* and *GetStretchBendParams*, have no meaning in the SMIRNOFF force field and SHOULD NOT be used.

The following methods are publicly inherited from *OEFFParams*:

- *CreateCopy*
- *GetCoulombParams*
- *GetStretchParams*
- *GetAtomTypeIndex*
- *GetEquilibriumAngle*
- *GetTorsionParams*
- *GetAtomTypeName*

- *GetEquilibriumBondLength*
- *GetVdwParams*
- *GetBendParams*
- *GetOutOfPlaneParams*
- *GetVdwRadius*
- *GetContactDistance*
- *GetStretchBendParams*
- *PrepMol*
- *GetVdwParams*
- *GetStretchParams*
- *GetBendParams*
- *GetTorsionParams*
- *GetOutOfPlaneParams*
- *HasVdwParams*
- *HasStretchParams*
- *HasBendParams*
- *HasTorsionParams*
- *GetTitle*

Constructors

```
OEFF14SBSmirnoffParams()
OEFF14SBSmirnoffParams(const OEFF14SBSmirnoffParams&)
```

Default and copy constructors.

Load

```
bool Load(const unsigned int fftype)
bool Load(const OEPlatform::oeisstream& ifs)
bool Load(const std::string& xmlFile)
```

Methods which load force field parameters for the SMIRNOFF/Parsley part of the potential. First method is used to load available build-in parameter set defined in the *OESmirnoffType* namespace. Second method takes the name of the parameter file in XML format. This file might contain user custom parameters. Third method takes a stream object oeisstream which contains Smirnoff parameters. It is a convenience method which might be used in user code when the parameters XML file is converted to a stream.

3.11.10 OEComplexEnergies

```
class OEComplexEnergies
```

This class provides energy components resulting from energy calculation for a protein-ligand complex.

The *OEComplexEnergies* class defines the following public methods:

- *GetComplexSolvationEnergy*
- *GetDesolvationEnergy*
- *GetHostComponents*
- *GetHostEnergy*
- *GetHostSolvationEnergy*
- *GetInterComponents*
- *GetInterEnergy*
- *GetLigandComponents*
- *GetLigandEnergy*
- *GetLigandSolvationEnergy*
- *GetTotalEnergy*

Constructors

```
OEComplexEnergies()
OEComplexEnergies(const OEComplexEnergies&)
```

Default and copy constructors.

GetComplexSolvationEnergy

```
double GetComplexSolvationEnergy() const
```

Returns the solvation energy of the protein-ligand complex.

GetDesolvationEnergy

```
double GetDesolvationEnergy() const
```

Returns the desolvation energy.

GetHostComponents

```
OESystem::OEIterBase<OEOpt::OEFComponent>* GetHostComponents() const
```

Returns the intra-molecular energy components of the host (protein) molecule.

GetHostEnergy

```
double GetHostEnergy() const
```

Returns the total intra-molecular energy of the host (protein) molecule.

GetHostSolvationEnergy

```
double GetHostSolvationEnergy() const
```

Returns the solvation energy of the host (protein) molecule.

GetInterComponents

```
OESystem::OEIterBase<OEOpt::OEFComponent>* GetInterComponents() const
```

Returns the inter-molecular energy components of the complex.

GetInterEnergy

```
double GetInterEnergy() const
```

Returns the total inter-molecular energy.

GetLigandComponents

```
OESystem::OEIterBase<OEOpt::OEFComponent>* GetLigandComponents() const
```

Returns the intra-molecular energy components of the ligand molecule.

GetLigandEnergy

```
double GetLigandEnergy() const
```

Returns the total intra-molecular energy of the ligand molecule.

GetLigandSolvationEnergy

```
double GetLigandSolvationEnergy() const
```

Returns the solvation energy of the ligand molecule.

GetTotalEnergy

```
double GetTotalEnergy() const
```

Returns the total energy of the complex.

3.11.11 OEComplexFF

> **Attention**
>
> This API is currently available in C++ and Python.

```
class OEComplexFF : public OEMolPotential::OEMolFunc2
```

The *OEComplexFF* is a generic class for force-fields of complexes. This class defines an interface for combining multiple *OEForceField* to easily construct a force field for a host/guest or protein/ligand complex. The *OEComplexFF* extends the *OEMolFunc2* class to allow this being a collection of multiple force fields.

The following methods are publicly inherited from *OEFunc0*:

- *operator()*
- *GetFComponents*

- *NumVar*

The following methods are publicly inherited from *OEFunc1*:

- *operator()*

The following methods are publicly inherited from *OEMolFunc*:

- *Clear*
- *ClearPredicates*
- *GetInteractions*
- *Setup*

The *OEComplexFF* class defines the following public methods:

- *AdaptHostCoords*
- *GetLigandFF*
- *GetVar*
- *Set*
- *SetHostFlex*
- *SetNonBondCutoff*
- *SetupHost*

The following classes derive from this class:

- *OEMMFFComplex*
- *OESmirnoffComplex*
- *OEMMFFAmberComplex*
- *OEFF14SBSmirnoffComplex*

Constructors

```
OEComplexFF(const OEMolPotential::OEForceField& ligandFF,
            const OEMolPotential::OEForceField& hostFF,
            const OEMolPotential::OEInterMolFunc2& interFF)
OEComplexFF(const OEComplexFF&)
```

Default and copy constructors.

Construction of a *OEComplexFF* requires a *OEForceField* for the guest or the ligand molecule, a *OEForceField* for the host or the protein molecule, and an *OEInterMolFunc2* for the inter-molecular interactions.

AdaptHostCoords

```
bool AdaptHostCoords(double* coords, const double* var) const
```

Takes a set of variables as the second argument and provides the corresponding host molecule coordinates into the first argument. This method is used to obtain optimized coordinates for the flexible host or protein molecule.

3.11.12 GetLigandFF

```
const OEMolPotential::OEForceField& GetLigandFF() const
```

Returns reference to the underlying forcefield used for small molecules.

3.11.13 GetVar

```
void GetVar(double* var, const double* coords)
```

Takes reference coordinates of the molecule, and transforms them into the variable type or coordinate system (`var`) that will be used during an optimization. The transfored variables here provides the combined host/guest coordinates, whenoptimization involves flexible host or protein molecule.

3.11.14 Set

```
bool Set(const OESystem::OEUnaryPredicate<OEChem::OEBondBase> &pred,
         unsigned int functype)
bool Set(const OESystem::OEUnaryPredicate<OEChem::OEAtomBase> &pred,
         unsigned int functype)
bool Set(const OESystem::OEBinaryPredicate<OEChem::OEAtomBase,
↪OEChem::OEAtomBase> &pred,
         unsigned int functype)
bool Set(const OESystem::OEBinaryPredicate<OEChem::OEBondBase,
↪OEChem::OEBondBase> &pred,
         unsigned int functype)
```

These methods assigns interaction-control predicates, and only applies to the guest or the ligand molecule. The first argument is the predicate type to be passed to the interaction component. The second argument is a value or a set of binary OR'ed values taken from the *OEFuncType* namespace which specifies the intended target for the predicate assignment.

The `Set` method passes on the interaction-control predicates to the appropriate force field components contained. Method returns `false` if predicate was not set to any of the components contained by the force field.

Set must be called before the *Setup* for it to be effective.

3.11.15 SetHostFlex

```
bool SetHostFlex(const OESystem::OEUnaryPredicate<OEChem::OEAtomBase>& pred)
```

Set predicate to define the flexible portion of the host or protein molecule. If no predicate is set to define flexibility, the host or protein is considered rigid.

SetHostFlex must be called before the *Setup* for it to be effective.

3.11.16 SetNonBondCutoff

```
bool SetNonBondCutoff(const double)
```

Sets non-bonded interactions cutoff. Setting a value of `0.0` or less assumes that all interactions should be included without any cutoff.

3.11.17 SetupHost

```
bool SetupHost(const OEChem::OEMolBase& host)
```

This method defines the interface for initializing the host in the *OEComplexFF*.

SetupHost must be called before calling *Setup* .

3.11.18 OEComplexFFParameter

> **Attention**
>
> This API is currently available in C++ and Python.

```
class OEComplexFFParameter : public OESystem::OEMultiParameter
↪<OEMolPotential::OEForceField>
```

The *OEComplexFFParameter* represents parameter that has a value of type *OEComplexFF*.

Following methods are publicly inherited from OEParameter:

- AddAlias and GetAliases
- AddDetail and GetDetail
- AddIllegalRange and GetIllegalRanges
- AddIllegalValue and GetIllegalValues
- AddLegalRange and GetLegalRanges
- AddStringDefault , GetStringDefault and GetStringDefaults
- AddStringValue , GetStringValue andw GetStringValues
- ClearDefaults
- ClearValues
- CreateCopy
- GetBrief and SetBrief
- GetHasDefault
- GetHasValue
- GetIsList and SetIsList
- GetKeyless and SetKeyless
- GetName and SetName
- GetOrderPriority and SetOrderPriority
- GetVisibility and SetVisibility
- IsLegalString
- IsSet and IsSetToString

Following methods are publicly inherited from OEMultiParameter:

- AddLegalEntry

- GetDefault and SetDefault
- GetSetting
- GetValue and SetValue

Constructors

```
OEComplexFFParameter()
OEComplexFFParameter(const std::string& name,
                     const bool customAllowed)
OEComplexFFParameter(const OEComplexFFParameter&)
```

Default and copy constructors.

Constructs an *OEComplexFFParameter* instance using the specified set of parameters. The second argument in the second constructor represents if a custom force field can be used.

3.11.19 OEForceFieldParameter

> **Attention**
>
> This API is currently available in C++ and Python.

```
class OEForceFieldParameter : public OESystem::OEMultiParameter
↪<OEMolPotential::OEForceField>
```

The *OEForceFieldParameter* represents parameter that has a value of type *OEForceField*.

Following methods are publicly inherited from OEParameter:

- AddAlias and GetAliases
- AddDetail and GetDetail
- AddIllegalRange and GetIllegalRanges
- AddIllegalValue and GetIllegalValues
- AddLegalRange and GetLegalRanges
- AddStringDefault , GetStringDefault and GetStringDefaults
- AddStringValue , GetStringValue and GetStringValues
- ClearDefaults
- ClearValues
- CreateCopy
- GetBrief and SetBrief
- GetHasDefault
- GetHasValue
- GetIsList and SetIsList
- GetKeyless and SetKeyless
- GetName and SetName

- GetOrderPriority and SetOrderPriority
- GetVisibility and SetVisibility
- IsLegalString
- IsSet and IsSetToString

Following methods are publicly inherited from OEMultiParameter:

- AddLegalEntry
- GetDefault and SetDefault
- GetSetting
- GetValue and SetValue

Constructors

```
OEForceFieldParameter()
OEForceFieldParameter(const std::string& name,
                      const bool customAllowed)
OEForceFieldParameter(const OEForceFieldParameter&)
```

Default and copy constructors.

Constructs an *OEForceFieldParameter* instance using the specified set of parameters. The second argument in the second constructor represents if a custom forcefield can be used.

3.11.20 OEMMFFIEFF

Attention

This API is currently available in C++ and Python.

```
class OEMMFFIEFF : public OEMolPotential::OEGenericFF
```

The *OEMMFFIEFF* class facilitates creation of an instance of a MMFFIEFF force field. The MMFFIEFF hybrid force field uses MMFF for intramolecular interactions and Amber for intermolecular interactions. In addition, a force field instance can be extended by adding externally defined potential functions.

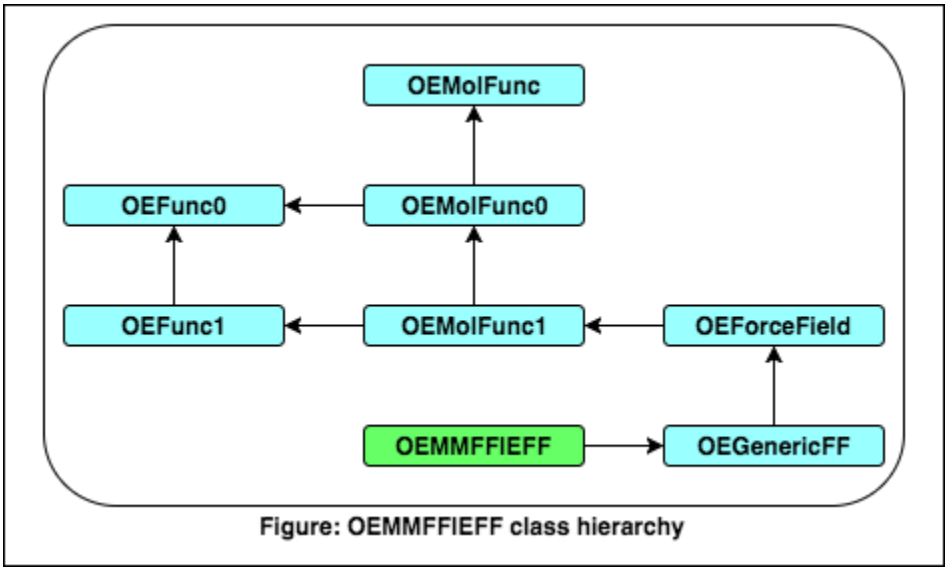


Figure: OEMMFFIEFF class hierarchy

The following methods are publicly inherited from *OEFunc0*:

- *operator()*
- *GetFComponents*
- *NumVar*

The following methods are publicly inherited from *OEFunc1*:

- *operator()*

The following methods are publicly inherited from *OEMolFunc*:

- *Clear*
- *ClearPredicates*
- *GetInteractions*
- *Setup*

The following methods are publicly inherited from *OEForceField*:

- *PrepMol*
- *Set*

The following methods are publicly inherited from *OEGenericFF*:

- *AddMolFunc*

- *GetComponents*
- *RemoveMolFunc*

Constructors

```
OEMMFFIEFF(const OEChem::OEMolBase& host)
OEMMFFIEFF(const OEChem::OEMolBase& host,
           const OEMMFFIEFFOptions& options)
```

Default and copy constructors.

The first constructor creates the force field with default options. The second constructor creates the force field with user specified options. The host refers to the host molecule for the interactions.

3.11.21 OEMMFFAmber

Attention

This API is currently available in C++ and Python.

```
class OEMMFFAmber : public OEMolPotential::OEGenericFF2
```

A hybrid force field that applies MMFF intramolecular terms to the ligand and Amber intramolecular terms to the host (protein), with Amber-based intermolecular interactions between them. This force field is intended for use in protein-ligand scoring and optimization workflows.

The host molecule must have 3D coordinates assigned. The ligand must have 3D coordinates and partial charges assigned (e.g. via `OEAssignCharges`) before calling *PrepMol*.

See also

- *OEMMFFAmberOptions* class
- *OEMMFFAmberComplex* class

The following methods are publicly inherited from *OEFunc0*:

- *operator()*
- *GetFComponents*
- *NumVar*

The following methods are publicly inherited from *OEFunc1*:

- *operator()*

The following methods are publicly inherited from *OEFunc2*:

- *operator()*

The following methods are publicly inherited from *OEMolFunc*:

- *Clear*
- *ClearPredicates*
- *GetInteractions*

- *Setup*

The following methods are publicly inherited from *OEForceField*:

- *PrepMol*
- *Set*

The following methods are publicly inherited from *OEGenericFF2*:

- *AddMolFunc*
- *GetComponents*
- *RemoveMolFunc*

The *OEMMFFAmber* class defines the following public methods:

- *PrepMol*
- *Setup*

Constructors

```
OEMMFFAmber(const OEChem::OEMolBase& host)
```

Constructs the force field using the given host molecule and default *OEMMFFAmberOptions*.

```
OEMMFFAmber(const OEChem::OEMolBase& host,
            const OEMMFFAmberOptions& options)
```

Constructs the force field using the given host molecule and the specified options.

PrepMol

```
bool PrepMol(OEChem::OEMolBase& mol, bool sweep = true,
             bool warnOK = true) const
```

Prepares a ligand molecule for use with this force field by assigning MMFF atom types and partial charges. Returns `true` on success.

sweep
: If `true`, sweeps through charge models to find one that covers all atoms.

warnOK
: If `true`, suppresses warnings for atoms that cannot be typed.

Setup

```
bool Setup(const OEChem::OEMolBase& mol)
```

Sets up the force field for the given ligand molecule. The molecule must have been prepared with *PrepMol* before calling this method. Returns `true` on success.

3.11.22 OEMMFFAmberComplex

> **Attention**
>
> This API is currently available in C++ and Python.

```
class OEMMFFAmberComplex : public OEComplexFF
```

The *OEMMFFAmberComplex* represents a force field for a protein-ligand complex system, where both the protein and ligand intra-molecular interactions are described with *OEMMFFParams* and the inter-molecular interactions are described with **Amber**.

The following methods are publicly inherited from *OEFunc0*:

- *operator()*
- *GetFComponents*
- *NumVar*

The following methods are publicly inherited from *OEFunc1*:

- *operator()*

The following methods are publicly inherited from *OEMolFunc*:

- *Clear*
- *ClearPredicates*
- *GetInteractions*
- *Setup*

The following methods are publicly inherited from *OEComplexFF*:

- *AdaptHostCoords*
- *GetVar*
- *Set*
- *SetHostFlex*
- *SetNonBondCutoff*
- *SetupHost*

Constructors

```
OEMMFFAmberComplex(const OEMolMMFF::OEMMFFParams& params = OEMolMMFF::OEMMFFParams(),
                   const OEMolPotential::OENonBondIntcsOptions& opts =␣
↪OEMolPotential::OENonBondIntcsOptions())
OEMMFFAmberComplex(const OEMMFFAmberComplex&)
```

Default and copy constructors.

3.11.23 OEMMFFAmberOptions

Attention

This API is currently available in C++ and Python.

```
class OEMMFFAmberOptions : public OESystem::OEOptions
```

Encapsulates the options that control the initialization of *OEMMFFAmber*.

See also

- *OEMMFFAmber* class

The *OEMMFFAmberOptions* class defines the following public methods:

- *GetAmberTerms*
- *GetIntermolecularVdWCutoff*
- *GetMMFFTerms*
- *GetUseExternalCharges*
- *GetUseMMFF94s*
- *SetAmberTerms*
- *SetIntermolecularVdWCutoff*
- *SetMMFFTerms*
- *SetUseExternalCharges*
- *SetUseMMFF94s*

Constructors

```
OEMMFFAmberOptions()
OEMMFFAmberOptions(const OEMMFFAmberOptions&)
```

Default and copy constructors.

CreateCopy

```
OEMMFFAmberOptions* CreateCopy() const
```

Returns a pointer to a copy of the options object.

GetAmberTerms

```
unsigned GetAmberTerms() const
```

Gets the constant defining which Amber force field terms to include.

GetIntermolecularVdWCutoff

```
double GetIntermolecularVdWCutoff() const
```

Gets the cutoff distance (in Angstroms) used for intermolecular van der Waals interactions.

GetMMFFTerms

```
unsigned GetMMFFTerms() const
```

Gets the constant defining which MMFF force field terms to include.

GetUseExternalCharges

```
bool GetUseExternalCharges() const
```

Gets the flag that controls whether partial charges already present on the molecule are used instead of MMFF-derived charges.

GetUseMMFF94s

```
bool GetUseMMFF94s() const
```

Gets the flag that controls whether the MMFF94s variant of parameters is used.

SetAmberTerms

```
bool SetAmberTerms(unsigned)
```

Sets the constant defining which Amber force field terms to include.

SetIntermolecularVdWCutoff

```
void SetIntermolecularVdWCutoff(double)
```

Sets the cutoff distance (in Angstroms) used for intermolecular van der Waals interactions.

SetMMFFTerms

```
bool SetMMFFTerms(unsigned)
```

Sets the constant defining which MMFF force field terms to include.

SetUseExternalCharges

```
void SetUseExternalCharges(bool)
```

Sets the flag that controls whether partial charges already present on the molecule are used instead of MMFF-derived charges.

SetUseMMFF94s

```
void SetUseMMFF94s(bool)
```

Sets the flag that controls whether the MMFF94s variant of parameters is used.

3.11.24 OEMMFFComplex

> **Attention**
>
> This API is currently available in C++ and Python.

```
class OEMMFFComplex : public OEComplexFF
```

The *OEMMFFComplex* provides a predefined interface to use *OEMMFF* for all interactions in a protein-ligand complex.

The following methods are publicly inherited from *OEFunc0*:

- *operator()*
- *GetFComponents*
- *NumVar*

The following methods are publicly inherited from *OEFunc1*:

- *operator()*

The following methods are publicly inherited from *OEMolFunc*:

- *Clear*
- *ClearPredicates*
- *GetInteractions*
- *Setup*

The following methods are publicly inherited from *OEComplexFF*:

- *AdaptHostCoords*
- *GetVar*
- *Set*
- *SetHostFlex*
- *SetNonBondCutoff*
- *SetupHost*

Constructors

```
OEMMFFComplex(const OEMolMMFF::OEMMFFParams& params = OEMolMMFF::OEMMFFParams(),
              const OEMolPotential::OENonBondIntcsOptions& opts =␣
↪OEMolPotential::OENonBondIntcsOptions())
OEMMFFComplex(const OEMMFFComplex&)
```

Default and copy constructors.

3.11.25 OEMMFFIEFFOptions

> **Attention**
>
> This API is currently available in C++ and Python.

```
class OEMMFFIEFFOptions : public OESystem::OEOptions
```

This class orovides interface to setup options required for *OEMMFFIEFF* initialization.

The *OEMMFFIEFFOptions* class defines the following public methods:

- *GetIEFFTerms*
- *GetMMFFTerms*
- *GetUseMMFF94s*
- *SetIEFFTerms*
- *SetMMFFTerms*
- *SetUseMMFF94s*

Constructors

```
OEMMFFIEFFOptions()
OEMMFFIEFFOptions(const OEMMFFIEFFOptions&)
```

Default and copy constructors.

GetIEFFTerms

```
unsigned int GetIEFFTerms() const
```

Gets the constant defining what terms of the IEFF force field to be used.

GetMMFFTerms

```
unsigned int GetMMFFTerms() const
```

Gets the constant defining what terms of the MMFF force field to be used.

GetUseMMFF94s

```
bool GetUseMMFF94s() const
```

Gets flag that defines if MMFF94s variation of parameters should be used.

SetIEFFTerms

```
bool SetIEFFTerms(const unsigned int)
```

Sets the constant defining what terms of the IEFF force field to be used.

SetMMFFTerms

```
bool SetMMFFTerms(const unsigned int)
```

Sets the constant defining what terms of the MMFF force field to be used.

SetUseMMFF94s

```
void SetUseMMFF94s(const bool)
```

Sets flag that defines if MMFF94s variation of parameters should be used.

3.11.26 OEMMFFSheffield

> **Attention**
>
> This API is currently available in C++ and Python.

```
class OEMMFFSheffield : public OEMolPotential::OEGenericFF2
```

The *OEMMFFSheffield* is a hybrid force field that combines the *OEMMFF* with the *OESheffield*. This force field is designed to be used exactly as that of *OEMMFF* or a variation of it, such as a truncated-OEMMFF, OEMMFF without electrostatics, or *OEMMFF* with *OESheffield*.

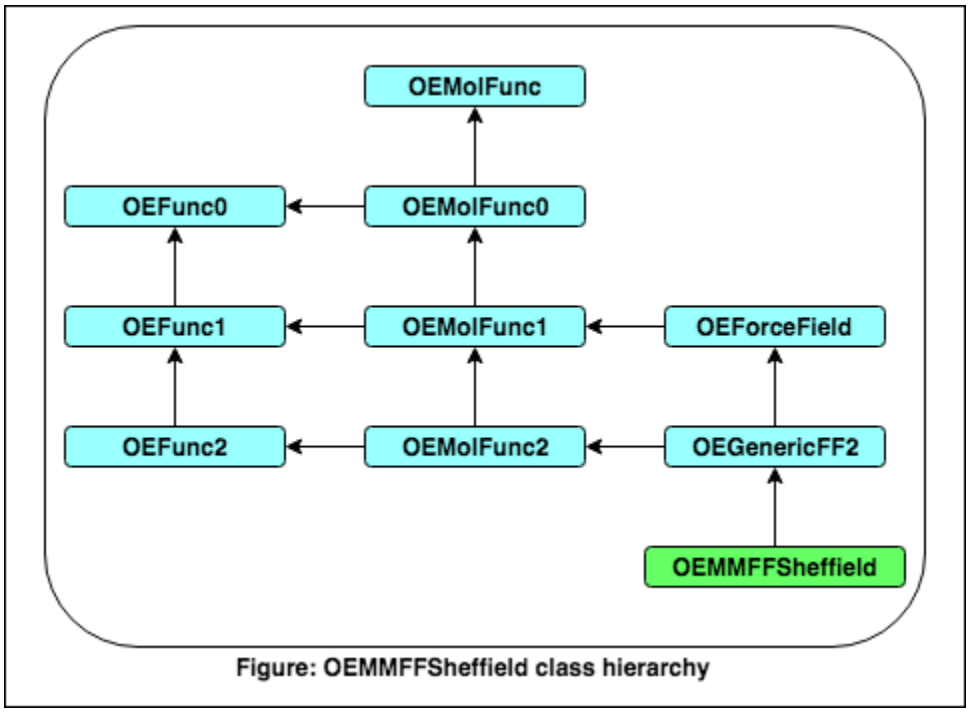


Figure: OEMMFFSheffield class hierarchy

> **See also**
>
> - *Intermolecular IEFF potential* section

- *MMFF* section

The following methods are publicly inherited from *OEFunc0*:

- *operator()*
- *GetFComponents*
- *NumVar*

The following methods are publicly inherited from *OEFunc1*:

- *operator()*

The following methods are publicly inherited from *OEFunc2*:

- *operator()*

The following methods are publicly inherited from *OEMolFunc*:

- *Clear*
- *ClearPredicates*
- *GetInteractions*
- *Setup*

The following methods are publicly inherited from *OEForceField*:

- *PrepMol*
- *Set*

The following methods are publicly inherited from *OEGenericFF2*:

- *AddMolFunc*
- *GetComponents*
- *RemoveMolFunc*

Constructors

```
OEMMFFSheffield(const OEMMFFSheffieldOptions& argOpts = OEMMFFSheffieldOptions())
```

Default constructor.

```
OEMMFFSheffield(const OEMMFFSheffieldOptions& argOpts = OEMMFFSheffieldOptions())
```

Creates the force field with user specified options.

3.11.27 OEMMFFSheffieldOptions

> **Attention**
>
> This API is currently available in C++ and Python.

```
class OEMMFFSheffieldOptions : public OESystem::OEOptions
```

This class provides interface to setup options required for *OEMMFFSheffield* initialization.

The *OEMMFFSheffieldOptions* class defines the following public methods:

- *GetDielectric* and *SetDielectric*
- *GetExponent* and *SetExponent*
- *GetForceFieldType* and *SetForceFieldType*

Constructors

```
OEMMFFSheffieldOptions()
OEMMFFSheffieldOptions(const OEMMFFSheffieldOptions&)
```

Default and copy constructors.

GetDielectric

```
double GetDielectric() const
```

See *SetDielectric* method.

GetExponent

```
double GetExponent() const
```

See *SetExponent* method.

GetForceFieldType

```
unsigned int GetForceFieldType() const
```

See *SetForceFieldType* method.

SetDielectric

```
bool SetDielectric(const double)
```

The **Dielectric** defines the solvent dielectric constant to be used for coulombic interactions with *OEMMFF*. **Default: 1.0.**

SetExponent

```
bool SetExponent(const double)
```

The **Exponent** defines the coulombic exponent term to be used for coulombic interactions with *OEMMFF*. **Default: 1.0.**

SetForceFieldType

```
bool SetForceFieldType(const unsigned int)
```

The **ForceFieldType** defines the variation of forcefield to be used. Options are defined in the *OEMMFFSheffieldFFType* namespace. **Default:** *OEMMFFSheffieldFFType.MMFF94Smod_NOESTAT*.

3.11.28 OESmirnoffComplex

> **Attention**
>
> This API is currently available in C++ and Python.

```
class OESmirnoffComplex : public OEComplexFF
```

The *OESmirnoffComplex* provides a predefined interface to use *OESmirnoff* for all interactions in a protein-ligand complex.

The following methods are publicly inherited from *OEFunc0*:

- *operator()*
- *GetFComponents*
- *NumVar*

The following methods are publicly inherited from *OEFunc1*:

- *operator()*

The following methods are publicly inherited from *OEMolFunc*:

- *Clear*
- *ClearPredicates*
- *GetInteractions*
- *Setup*

The following methods are publicly inherited from *OEComplexFF*:

- *AdaptHostCoords*
- *GetVar*
- *Set*
- *SetHostFlex*
- *SetNonBondCutoff*
- *SetupHost*

Constructors

```
OESmirnoffComplex(const OEMolSmirnoff::OESmirnoffParams& params =␣
↪OEMolSmirnoff::OESmirnoffParams(),
                  const OEMolPotential::OENonBondIntcsOptions& opts =␣
↪OEMolPotential::OENonBondIntcsOptions())
OESmirnoffComplex(const OESmirnoffComplex&)
```

Default and copy constructors.

3.12 OEFF Functions

3.12.1 OEClearForceFieldDummyAtom

```
void OEClearForceFieldDummyAtom(OEChem::OEAtomBase* atom)
```

Clear any force field dummy atom information associated with the specified atom by *OESetForceFieldDummyAtom*.

See also

- *OESetForceFieldDummyAtom* function
- *OEIsForceFieldDummyAtom* function
- *OEClearForceFieldDummyAtoms* function

Note

This function is defined in the **OEChem TK**, but in the OEFF main namespace.

3.12.2 OEClearForceFieldDummyAtoms

```
void OEClearForceFieldDummyAtoms(OEChem::OEMolBase& mol)
```

Clear any force field dummy atom information for all atoms in the molecule.

See also

- *OESetForceFieldDummyAtom* function
- *OEIsForceFieldDummyAtom* function
- *OEClearForceFieldDummyAtom* function

Note

This function is defined in the **OEChem TK**, but in the OEFF main namespace.

3.12.3 OEGetMMFFSheffieldFFName

```
std::string OEGetMMFFSheffieldFFName(const unsigned int)
```

Get the *OEMMFFSheffield* forcefield name, corresponding to the given type ID. The valid type ID's are defined in the *OEMMFFSheffieldFFType* namespace.

See also

- *OEGetMMFFSheffieldFFType* function

- *OEIsValidMMFFSheffieldFF* function

3.12.4 OEGetMMFFSheffieldFFType

```
unsigned int OEGetMMFFSheffieldFFType(const std::string&)
```

Get the *OEMMFFSheffield* forcefield type ID, as defined in the *OEMMFFSheffieldFFType* namespace, corresponding to the given string.

See also

- *OEGetMMFFSheffieldFFType* function
- *OEIsValidMMFFSheffieldFF* function

3.12.5 OEIsForceFieldDummyAtom

```
bool OEIsForceFieldDummyAtom(const OEChem::OEAtomBase* atom)
```

Returns whether *OESetForceFieldDummyAtom* has been used on the atom to indicate it can be used as a force field dummy atom. The atom's atomic number does *not* have to be `OEElemNo.Du` for the function to return `true`.

See also

- *OESetForceFieldDummyAtom* function
- *OEClearForceFieldDummyAtom* function
- *OEClearForceFieldDummyAtoms* function

Note

This function is defined in the **OEChem TK**, but in the OEFF main namespace.

3.12.6 OEIsValidMMFFSheffieldFF

```
bool OEIsValidMMFFSheffieldFF(const std::string&)
bool OEIsValidMMFFSheffieldFF(const unsigned int)
```

Check if the specified value correspond to a valid *OEMMFFSheffield* forcefield.

See also

- *OEGetMMFFSheffieldFFName* function
- *OEGetMMFFSheffieldFFType* function

3.12.7 OESetForceFieldDummyAtom

```
void OESetForceFieldDummyAtom(OEChem::OEAtomBase* atom)
```

Modify information associated with an atom to indicate that if the atom has atomic number `OEElemNo.Du` it can be used for certain force field calculations where it would not normally be valid.

Note

Currently, only **MMFF** and **Amber** support support dummy atoms at all, and even then *only* for calculations involving non-bonded terms and when this special indication has been set.

See also

- *OEIsForceFieldDummyAtom* function
- *OEClearForceFieldDummyAtom* function
- *OEClearForceFieldDummyAtoms* function

Note

This function is defined in the **OEChem TK**, but in the OEFF main namespace.

3.12.8 OEFFGetArch

```
const char* OEFFGetArch()
```

Returns the internal build string used by OpenEye, Cadence Molecular Sciences to identify the hardware architecture. The format of these strings may change over time.

3.12.9 OEFFGetLicensee

```
std::string OEFFGetLicensee()
bool OEFFGetLicensee(std::string& licensee)
```

Returns the licensee name from the license file.

3.12.10 OEFFGetPlatform

```
const char* OEFFGetPlatform()
```

Returns the internal build string used by OpenEye, Cadence Molecular Sciences to identify a platform. The format of these strings may change over time, and future distributions may contain different values even when using the same operating system, compiler and processor. For example, on a `x86_64` Red Hat Enterprise Linux box this would return `redhat-RHEL7-g++4.1-x64`.

3.12.11 OEFFGetRelease

```
const char* OEFFGetRelease()
```

Returns the release name of the *OEFF* library being used. This returns a value similar to `2.0.0` for production versions of the library, and `2.0.0 debug` for the checking version of the library.

3.12.12 OEFFGetSite

```
std::string OEFFGetSite()
bool OEFFGetSite(std::string& site)
```

Returns the physical site location of the licensee as registered in the license file.

3.12.13 OEFFGetVersion

```
unsigned int OEFFGetVersion()
```

Returns the version number of the library being used. This is an unsigned integer value indicating the date on which the library was built, for example `20141001`, for the 1st of October 2014. This value should be used when reporting problems, and unlike the release string, may be used in comparisons if needed.

3.12.14 OEFFIsLicensed

```
bool OEFFIsLicensed(const char* feature = 0, unsigned int* expdate = 0)
```

Determines whether a valid license file is present. This function may be called without a legitimate run-time license to determine whether it is safe to call any **OEFF TK** functionality.

The 'feature' argument can be used to check for a valid license to **OEFF TK** along with the following features: 'python', 'java', or 'clr' (for CSharp).

The 'expdate' argument can be used to obtain the expiration date of the current ***OEFF TK** license.

See also

- `OEChemIsLicensed` function

3.13 OEFF Constants

3.13.1 OELigandFFType

This namespace contains constants that define force fields type appropriate for working with ligands.

MMFF94

The Merck Molecular Force Field (MMFF). Name: mmff94

MMFF94S

The 94s variation of the Merck Molecular Force Field (MMFF). Name: mmff94s

PARSLEY

The latest version (1.3.1) of the Parsley series of force field from the OPENFF consortium. Name: parsley

SAGE

The latest version (2.0.0) of the Sage series of force field from the OPENFF consortium. Name: sage

3.13.2 OEMMFFSheffieldFFType

This namespace contains constants that are used with *OEMMFFSheffieldOptions* to define a specific *OEMMFFSheffield* forcefield.

MMFF

The Merck Molecular Force Field (MMFF). Name: mmff

MMFF_NOESTAT

The MMFF forcefield without electrostatics. Name: mmff_NoEstat

MMFF_TRUNC

The MMFF forcefield truncated at potential minimum. Name: mmff_Trunc

MMFF_SHEFF

The MMFF forcefield with Sheffield solvation. Name: mmff_Sheff

MMFF94S

The 94s variation of the Merck Molecular Force Field (MMFF). Name: mmff94s

MMFF94S_NOESTAT

The MMFF94S forcefield without electrostatics. Name: mmff94s_NoEstat

MMFF94S_TRUNC

The MMFF94S forcefield truncated at potential minimum. Name: mmff94s_Trunc

MMFF94S_SHEFF

The MMFF94S forcefield with Sheffield solvation. Name: mmff94s_Sheff

MMFF94Smod

The modified 94s variation of the Merck Molecular Force Field (MMFF). Name: mmff94smod

MMFF94Smod_NOESTAT

The modified MMFF94S forcefield without electrostatics. Name: mmff94smod_NoEstat

MMFF94Smod_TRUNC

The modified MMFF94S forcefield truncated at potential minimum. Name: mmff94smod_Trunc

MMFF94Smod_SHEFF

The modified MMFF94S forcefield with Sheffield solvation. Name: mmff94smod_Sheff

Invalid

Invalid force field type.

CHAPTER

FOUR

RELEASE HISTORY

4.1 OEFF TK 2.9.2

4.1.1 Minor Bug Fixes

- The *OEQuatAdaptor* now passes the coordinates to the internal *OEMolFunc* when appropriate.

4.2 OEFF TK 2.9.1

Minor internal improvements have been made.

4.3 OEFF TK 2.9.0

4.3.1 New Features

- The new method *GetLigandFF* has been added to *OEComplexFF* that enables access to the underlying ligand force field.
- The following new classes have been introduced to enable accessing data associated with any *OESmirnoffParams* parameters:
  - *OESmirnoffData*
  - *OESmirnoffStretchData*
  - *OESmirnoffBendData*
  - *OESmirnoffTorsionData*
  - *OESmirnoffImproperData*
  - *OESmirnoffVdWData*
- The following new methods have been added to *OESmirnoffParams* to enable accessing data associated with various parameters:
  - *AddBendData*
  - *AddImproperData*
  - *AddStretchData*
  - *AddTorsionData*
  - *AddVdWData*
  - *GetBendData*

  - *GetImproperData*
  - *GetStretchData*
  - *GetTorsionData*
  - *GetVdWData*
  - *SetBendData*
  - *SetImproperData*
  - *SetStretchData*
  - *SetTorsionData*
  - *SetVdWData*

- The function *OEIsValidMMFFMolecule* has a new overload that allows it to ignore charges while checking if MMFF parameters are available for the specified molecule.

4.3.2 Minor Bug Fixes

- An issue that was causing an occasional segmentation fault in the *SetHostFlex* method in the *OEFF14SBSageComplex* class has been fixed.
- The method *SetupConf* in *OEMolFunc* is now properly wrapped in Python.

4.4 OEFF TK 2.8.0

4.4.1 New features

- The constant *OESmirnoffType.SAGE_OPENFF* now refers to the 2.2.1 version of the Sage OpenFF force field.
- Any SMIRNOFF set of parameters loaded using *OESmirnoffParams*, including the built-in *OESageParams* and *OEParsleyParams*, are now supplemented with van der Waals parameters for many metals and counterions, as used with the `OpenMM` package.
- The *OEMMFF94sParams* and *OEMMFFParams* are now supplemented with a generic set of van der Waals parameters for many metals and counterions to improve robustness.
- A new function, *OEIsValidMMFFMolecule*, has been added that can be used to check if *OEMMFFParams* are available for a molecule.

4.4.2 Minor Bug Fixes

- Loading a SMIRNOFF set of parameters using *OESmirnoffParams* no longer picks up the library charges.

4.5 OEFF TK 2.7.1.3

- The constant *OESmirnoffType.SAGE_OPENFF* now refers to the 2.2.0 version of the Sage-OpenFF force field.

4.6 OEFF TK 2.7.1

4.6.1 New features

- The following preliminary API classes have been made permanent.
  - *OENonBondIntcsOptions*

- *OEComplexFF*
- *OEComplexFFParameter*
- *OEFF14SBParsley*
- *OEFF14SBParsleyComplex*
- *OEFF14SBSage*
- *OEFF14SBSageComplex*
- *OEFF14SBSmirnoff*
- *OEFF14SBSmirnoffComplex*
- *OEForceFieldParameter*
- *OEMMFFAmberComplex*
- *OEMMFFComplex*
- *OESmirnoffComplex*

4.6.2 Minor bug fixes

- The `Get` and `Set` methods in *OEInteractParams* are now exposed properly in Python.

4.7 OEFF TK 2.7.0

4.7.1 New features

- A new method, *HasInternalCharges*, has been added to *OEForceField* that returns a boolean indicating whether the parameters in the *OEForceField* contain charges.
- A new method, *HasCoulomb*, has been added to *OEForceField* that returns a boolean indicating whether coulombic interactions are turned on.
- A new method, *GetNonBondOptions*, has been added to *OENonBondPotBase* that provides access to its internal *OENonBondIntcsOptions*.
- A new method, *SetNonBondCutoff*, has been added to *OEComplexFF* that provides a way to set nonbonded interactions cutoffs.

4.7.2 Major bug fixes

- The internal algorithms for VdW parameter assignment with *OESmirnoffParams* have been modified to improve performance in working with large molecules.

4.7.3 Minor bug fixes

- The `GetTitle` method for *OESmirnoff* now properly returns the user set title of the force field.

4.8 OEFF TK 2.5.2.1

- Support for the latest released version of the force field from the Open Force Field Initiative (Sage 2.1.0) has been added. The constant `OESmirnoffType.SAGE_OPENFF` now refers to this latest version of Sage force field.

4.9 OEFF TK 2.5.2.0

4.9.1 Major bug fixes

- The *OEComplexFF* class is now derived from *OEMolFunc2* and has limited support for second derivatives when the protein is fixed.

4.10 OEFF TK 2.5.1

4.10.1 New features

- Two new constants, *OESmirnoffType.PARSLEY_OPENFF* and *OESmirnoffType.SAGE_OPENFF*, have been added to *OESmirnoffType* that define the latest versions of `Parsley` and `Sage` forcefields respectively.

4.10.2 Major bug fixes

- An issue resulting in a crash when a combine force field ff14SB/OpenFF_Sage or ff14SB/OpenFF_Parsley was applied to a small molecule which was not a part of the protein-ligand complex has been fixed.
- VdW parameters for atoms which are not explicitly given in the Sage or Parsley versions of OpenFF and are not tabulated in ff14SB parameter sets are now taken from the original AMBER parameter set. An example of such systems are proteins with metal cofactors. Until now such systems failed to be processed by instances of force field classes *OEFF14SBParsley* or *OEFF14SBSage*.

4.10.3 Minor bug fixes

- A missing improper torsion parameter in ff14SB force field for residue PTR, which is defined by three carbon atoms from a benzene ring and oxygen from a phosphate group, has been added.

4.11 OEFF TK 2.5.0

4.11.1 New features

- Class *OESheffield* has been extended with a model which approximates solvation of proteins. This extension might be useful for evaluation of differences in solvation energies between protein conformations and in protein loop modeling.
- The following methods have been added to the *OEComplexEnergies* class:
  - *GetComplexSolvationEnergy*
  - *GetDesolvationEnergy*
  - *GetHostSolvationEnergy*
  - *GetLigandSolvationEnergy*
- A new *OELigandFFType* namespace has been added that defines force fields suitable for working with ligands.
- A new *OESheffieldParamType* namespace has been added that defines different available Sheffield parameter types.
- Two new methods have been added to *OESheffieldOptions*:
  - *GetC*
  - *SetC*

4.11.2 Major bug fixes

- An issue have been fixed where failure to assign atom vdw parameters were ignored resulting in calculations being performed with values of zero being assigned as the parameters. Effect of this fix could be observed in calculations performed using the following force fields:
  - *OESmirnoff*
  - *OEParsley*
  - *OESage*
  - *OEFF14SBSmirnoff*
  - *OEFF14SBParsley*
  - *OEFF14SBSage*
- Semiempirical AM1/PM3 calculations are not allowed for molecules in which atoms indexing is not compatible with the number of atoms. Until now passing molecules with that feature could caused segmentation fault.

4.12 OEFF TK 2.4.0

Fall 2021

4.12.1 New features

- The following new classes have been added to provide support for the OpenFF 2.0.0 Sage force field:
  - *OESageParams*
  - *OEFF14SBSageParams*
  - *OESage*
  - *OEFF14SBSage*
  - *OEFF14SBSageComplex*
- *OEForceFieldParameter* and *OEComplexFFParameter* now accept case-insensitive string input values.

4.12.2 Minor bug fixes

- A potential risk of vectors going out-of-bounds while working with *OESmirnoff* and related classes have been addressed.

4.13 OEFF TK 2.3.1

July 2021

Minor internal improvements have been made.

4.14 OEFF TK 2.3.0

Fall 2020

4.14.1 New features

- A new generic set of force field potential classes has been added to simplify implementing new force fields in OEFF TK:
  - *OEFFPotential*
  - *OEStretchPotential*
  - *OEBendPotential*
  - *OETorsionPotential*
  - *OEOutOfPlanePotential*
  - *OENonBondPotBase*
  - *OENonBondPotential*
  - *OEVdwPotential*
  - *OECoulombPotential*
  - *OEInterNonBondPotBase*
  - *OEInterNonBondPotential*
  - *OEInterVdwPotential*
  - *OEInterCoulombPotential*
- With the introduction of the new generic force field potential classes in **OEFF TK**, several previously existing classes have been deprecated. The following table shows the deprecated classes and their replacements:

Deprecated Class	New Class
`OEMolPotential::OEStretchBase`	*OEStretchPotential*
`OEMolMMFF::OEMMFFStretch`	
`OEMolSmirnoff::OESmirnoffStretch`	
`OEMolMMFF::OEMMFFBend`	*OEBendPotential*
`OEMolSmirnoff::OESmirnoffBend`	
`OEMolMMFF::OEMMFFTorsion`	*OETorsionPotential*
`OEMolSmirnoff::OESmirnoffTorsion`	
`OEMolSmirnoff::OESmirnoffOutOfPlane`	*OEOutOfPlanePotential*
`OEMolPotential::OENonBondBase`	*OENonBondPotBase*
`OEMolMMFF::OEMMFFVdw`	*OEVdwPotential*
`OEMolSmirnoff::OESmirnoffVdw`	
`OEMolMMFF::OEMMFFCoulomb`	*OECoulombPotential*
`OEMolSmirnoff::OESmirnoffCoulomb`	
`OEMolMMFF::OEMMFFInterVdw`	*OEInterVdwPotential*
`OEMolMMFF::OEMMFFInterVdwNN`	
`OEMolSmirnoff::OESmirnoffInterVdw`	
`OEMolMMFF::OEMMFFInterCoulomb`	*OEInterCoulombPotential*

- New OpenFF parameters `parsley_1.2.1` have been added as built-in parameters and can be used in the construction of an *OESmirnoff* object.
- Two new classes, *OEParsleyParams* and *OEParsley*, have been added that work with the latest versions of available OpenFF Parsley force fields.
- New FF14SB ([Maier-2015]) protein force field parameters have been implemented. These parameters are accessible through the following parameter and force field classes.

  - *OEFF14SBSmirnoffParams*: an FF14SB for proteins with custom *OESmirnoffParams* for cofactor and/or ligand
  - *OEFF14SBParsleyParams*: an FF14SB for proteins with *OEParsleyParams* for cofactor and/or ligand
  - *OEFF14SBSmirnoff*: a force field for proteins with *OEFF14SBSmirnoffParams*
  - *OEFF14SBParsley*: a force field for proteins with *OEFF14SBParsleyParams*
  - *OEFF14SBSmirnoffComplex*: a force field for protein-ligand complexes with *OEFF14SBSmirnoffParams*
  - *OEFF14SBParsleyComplex*: a force field for protein-ligand complex with *OEFF14SBParsleyParams*
- A new class, *OEMMFFAmberComplex*, has been added that allows protein-ligand optimization with protein flexibility. Accordingly, `OEFF::OEMMFFAmber` has been deprecated.
- A new preliminary class, *OEComplexFFParameter*, has been added as part of the extended set of OEParameter classes. The new parameter class works with predefined complex force fields using an *OEComplexFF* object.
- The methods `OEFF::OEComplexFF::PrepMol` and `OEFF::OEComplexFF::PrepHost` have been removed from the preliminary API *OEComplexFF* as they are no longer required to setup an *OEComplexFF*.
- A new class, *OEComplexEnergies*, has been added that represents energy components of a protein-ligand complex.
- A new overload of `GetFComponents` has been added that provides the energy components in the form of an *OEComplexEnergies*.

4.14.2 Documentation changes

- All the **OEFF TK** examples have been reorganized to reflect the modified API behavior.

4.15 OEFF TK 2.2.0

4.15.1 New features

- OpenFF parameters `parsley_1.0.0` and `parsley_1.1.1` are now built-in parameters and can be used in the construction of an *OESmirnoff* object.
- Beginning with this release, custom OpenFF parameters in XML file format can be used. This functionality is achieved by the overloaded method `OESmirnoffParams.Load`.
- A new set of preliminary api force field classes has been added to simplify working with the host-guest complex system:
  - *OEComplexFF*
  - *OEMMFFComplex*
  - *OESmirnoffComplex*
- A new preliminary class, `OEMolMMFF::OEMMFFNonBond`, that combines `OEMolMMFF::OEMMFFVdw` and `OEMolMMFF::OEMMFFCoulomb` has been added.
- A new preliminary class, `OEMolMMFF::OEMMFFInterNonBond`, that combines `OEMolMMFF::OEMMFFInterVdw` and `OEMolMMFF::OEMMFFInterCoulomb` has been added.
- A new preliminary class, `OEMolSmirnoff::OESmirnoffNonBond`, that combines `OEMolSmirnoff::OESmirnoffVdw` and `OEMolSmirnoff::OESmirnoffCoulomb` has been added.
- A new preliminary class, `OEMolSmirnoff::OESmirnoffInterCoulomb`, has been added to use with the intermolecular coulomb component in SMIRNOFF-based force fields.

- A new preliminary class, `OEMolSmirnoff::OESmirnoffInterNonBond`, that combines `OEMolMMFF::OESmirnoffInterVdw` and `OEMolSmirnoff::OESmirnoffInterCoulomb` has been added.
- A new preliminary class, *OENonBondIntcsOptions*, has been added that provides consistency in setting non-bond interaction calculations in various force fields.
- New constructors for *OEMMFF* and *OESmirnoff* have been added that provide consistency and use *OENonBondIntcsOptions* as arguments.
- Two new SMIRNOFF force fields from the Open Force Field Initiative, `parsley_1.0.0` and `parsley_1.1.1`, have been added as built-in force fields. They are defined by types PARSLEY_OPENFF100 and PARSLEY_OPENFF111, respectively.
- A new preliminary class, *OEForceFieldParameter*, has been added as part of the extended set of OEParameter classes. The new parameter class works with predefined force fields and custom SMIRNOFF force fields, as well as user-defined *OEForceField* force fields using *OEForceField* for the same parameter object.
- Two new public methods have been added to *OEVdwParams*:
  - *GetAtom1*
  - *GetAtom2*
- New public methods have been added to *OEFFParams*:
  - *HasVdwParams*
  - *HasStretchParams*
  - *HasBendParams*
  - *HasTorsionParams*
  - *GetTitle*

4.15.2 Major bug fixes

- Copy constructors and assignment operators in the classes OESmirnoffStretch, OESmirnoffBend, OESmirnoffTorsion, and OESmirnoffOutOfPlane had previously failed to copy their private member object of the *OESmirnoffParams* type. This has been fixed.

  For updated information, see *Deprecated OEFF Classes*.

4.15.3 Minor bug fixes

- The following methods now return `bool` instead of a `void`:
  - *OEMMFFSheffieldOptions.SetDielectric*
  - *OEMMFFSheffieldOptions.SetExponent*
  - *OEMMFFSheffieldOptions.SetForceFieldType*
  - *OESheffieldOptions.SetA*
  - *OESheffieldOptions.SetB*
  - *OESheffieldOptions.SetSoluteDielectricConstant*
  - *OESheffieldOptions.SetSolventDielectricConstant*
- The names of the potential components `Smirnoff Angle` and `Smirnoff Imp Torsion` were changed to `Smirnoff Bend` and `Smirnoff Improper Torsion`, respectively, in order to be compatible with the same term names in the `MMFF94` force field.

- The numerical stability in the calculation of second derivatives in the OESmirnoffOutOfPlane class has been improved.

4.16 OEFF TK 2.1.0

4.16.1 New features

- A new class, *OESmirnoff*, has been added that facilitates creating a SMIRNOFF force field (release 0.3), a small-molecule force field that uses extended SMARTS patterns for its chemical representation of parameters (https://openforcefield.org/). The following classes have also been added that define interaction functions using the functional forms in the SMIRNOFF force field:
  - *OESmirnoffParams*
  - `OEMolSmirnoff::OESmirnoffStretch`
  - `OEMolSmirnoff::OESmirnoffBend`
  - `OEMolSmirnoff::OESmirnoffTorsion`
  - `OEMolSmirnoff::OESmirnoffOutOfPlane`
  - `OEMolSmirnoff::OESmirnoffVdw`
  - `OEMolSmirnoff::OESmirnoffInterVdw`
  - `OEMolSmirnoff::OESmirnoffCoulomb`
- The following classes, derived from the parent class *OEInteractParams*, have been added to create a set of force field parameters for a given molecule:
  - *OEStretchParams*
  - *OEBendParams*
  - *OETorsionParams*
  - *OEVdwParams*
  - *OEOutOfPlaneParams*
- The `OEFFParam` namespace has been expanded to support SMIRNOFF force field torsion potential.
- A new method, `GetIterLimit`, has been added that allows assessing the state of an *OEOptimizer0*.
- Two new methods, `GetIterLimit` and `GetTolerance`, have been added that allow assessing the state of an *OEOptimizer1*.
- Two new methods, `GetIterLimit` and `GetTolerance`, have been added that allow assessing the state of an *OEOptimizer2*.

4.16.2 Major bug fixes

- A memory leak in *OEInterAdaptor* has been fixed.
- A memory leak in OEAM1 has been fixed.

4.16.3 Documentation changes

- Theory documentation for SMIRNOFF has been added (see *SMIRNOFF*).
- A new example program was written that demonstrated calculating energy and optimizing a single ligand in vacuum.

- The following example programs have been corrected:
  - section_mmff_example
  - section_mmff_energy_example
  - section_mmff_interactions_example
  - section_mmff_complex_example
  - section_mmff_amber_example
  - section_mmff_subset_example
  - section_mmff_tor_example
  - section_mmff_amber_quat_example

4.17 OEFF TK 2.0.4

- Minor internal improvements have been made.

4.18 OEFF TK 2.0.3

4.18.1 New features

- A new *OEMMFFSheffield* class and a new *OEMMFFSheffieldFFType* namespace have been added to the force field suites for molecular interactions. This is a hybrid force field consisting of *OEMMFF* and *OESheffield*.
- A new *OEMMFFSheffieldOptions* class has been added to set all options of an *OEMMFFSheffield* object during its construction.
- A new *OEGetMMFFSheffieldFFName* function has been added to provide the force field name corresponding to a specific variation of a force field type.
- A new *OEGetMMFFSheffieldFFType* function has been added to provide a specific variation of the force field type corresponding to a force field name.
- A new *OEIsValidMMFFSheffieldFF* function has been added that can check if a type ID or name corresponds to a valid *OEMMFFSheffield* force field type.
- New *OEForceField.GetTitle* and *OEForceField.SetTitle* methods have been added to set and return the title of the force field.

4.18.2 Minor bug fixes

- Water is now treated as natural residue in the AMBER force field.

4.18.3 Documentation changes

- A minor fix has been made in the example in the *MMFF interactions of a single ligand* section.

4.19 OEFF TK 2.0.2

- Minor internal improvements have been made.

4.20 OEFF TK 2.0.1

4.20.1 New features

- A set of classes that were previously derived from *OEMolFunc1* are now derived from *OEMolFunc2*. With this change, all of these classes now provide analytical Hessians and can be optimized with *OENewtonOpt*. The list of classes is as follows:
  - `OEMolMMFF::OEMMFFBend`
  - `OEMolMMFF::OEMMFFStretch`
  - `OEMolMMFF::OEMMFFTorsion`
  - `OEMolMMFF::OEMMFFCoulomb`
  - `OEMolMMFF::OEMMFFOutOfPlane`
  - *OEMMFFStretchBend*
  - `OEMolMMFF::OEMMFFVdw`
  - *OESheffield*
- A new class, *OEGenericFF2*, has been added that extends *OEForceField* to be an *OEMolFunc2* and provide Hessians.
- Two of the *OEForceField* classes, *OEMMFF* and `OEFF::OEMMFFAmber`, now derive from *OEGenericFF2*.
- Two of the *OEForceField* methods, `OEMolPotential::OEForceField::Add` and `OEMolPotential::OEForceField::Remove`, have been deprecated and replaced by more user-friendly methods in the derived classes *OEGenericFF* and *OEGenericFF2*.

  The following table shows the older, deprecated functions and their replacements:

Deprecated	New
`OEMolPotential::OEForceField::Add`	*OEGenericFF.AddMolFunc* *OEGenericFF2.AddMolFunc*
`OEMolPotential::OEForceField::Remo`	*OEGenericFF.RemoveMolFunc* *OEGenericFF2.RemoveMolFunc*

- A new method, *OEGenericFF.GetComponents*, has been added that gives access to the function components of the force field.
- A new class, *OEMolAdaptor2*, has been added that extends the *OEMolAdaptor* to be an *OEMolFunc2* and provide Hessians.
- The *OESubsetAdaptor* class now derives from *OEMolAdaptor2*.
- A new class, *OENumericMolFunc2*, has been added that can be used to convert any *OEMolFunc1* into an *OEMolFunc2* with numerical Hessians.
- A new class, *OEScaledMolFunc*, has been added that can be used to scale (up or down) any *OEMolFunc1* or *OEMolFunc2* derived classes.

4.20.2 Minor Bug Fixes

- The constructor for the OEMMFFCoulomb class previously had an unused *OEFFParams* parameter. A new overload without this parameter has been added, and the older constructor has been deprecated. For updated information, see *Deprecated OEFF Classes*.

4.21 OEFF TK 2.0.0

4.21.1 Licensing

- **OEFF TK** is available to anyone with an **OESzybki TK** license. Visit https://www.eyesopen.com/contact if you need an **OESzybki TK** license.

4.21.2 OEFF functionality

This is the first official release of **OEFF TK**. The underlying functionality was previously released in C++ as part of the **OESzybki TK** advanced interface. This toolkit is currently available in C++ and Python. An overview of its capabilities is provided below.

OEFF TK provides access to powerful low-level functionality for advanced users to develop molecular modeling applications. With **OEFF TK**, users also have the ability to create variations of molecular interaction classes for energy evaluation and optimization, adding to the built-in interactions, if desired. **OESzybki TK** complements **OEFF TK** as the high-level API for molecular modeling functionalities.

4.21.3 Objective functions and molecular interactions

All molecular interactions are objective functions with additional API points for molecular definitions. A set of pure virtual classes that can be used to define new interactions are provided for both generic objective functions and molecule specific interaction functions. The following virtual base classes are available:

- *OEFunc0*
- *OEFunc1*
- *OEFunc2*
- *OEMolFunc*
- *OEMolFunc0*
- *OEMolFunc1*
- *OEMolFunc2*

4.21.4 Force fields

Force fields and components of force fields are defined as molecule objective functions and are derived from the above-specified base classes. The following force fields and components are available:

- *OEMMFF*
- `OEMolMMFF::OEMMFFBend`
- `OEMolMMFF::OEMMFFStretch`
- *OEMMFFStretchBend*
- `OEMolMMFF::OEMMFFOutOfPlane`
- `OEMolMMFF::OEMMFFTorsion`
- `OEMolMMFF::OEMMFFVdw`
- `OEMolMMFF::OEMMFFInterVdw`
- `OEMolMMFF::OEMMFFInterVdwNN`
- `OEMolMMFF::OEMMFFCoulomb`
- `OEMolMMFF::OEMMFFInterCoulomb`

- `OEFF::OEMMFFAmber`
- *OEMMFFIEFF*
- *OESheffield*

4.21.5 Optimizers

A powerful set of optimizers that can be used to optimize objective functions is now available:

- *OEBFGSOpt*
- *OEConjGradOpt*
- *OEDFPOpt*
- *OEFletcherReevesOpt*
- *OENewtonOpt*
- *OERibierePolakOpt*
- *OESteepestDescentOpt*

4.21.6 Adaptors

Adaptors that can be used to optimize molecule objective functions in transformed coordinate space are now available:

- *OEInterAdaptor*
- *OEQuatAdaptor*
- *OESubsetAdaptor*
- *OETorAdaptor*
- *OETorQuatAdaptor*

4.21.7 New features

- Two new intermolecular force fields, `OEFF::OEMMFFAmber` and *OEMMFFIEFF*, have been added to the force field suites for molecular interactions. Both force fields use *OEMMFF* for intermolecular interactions and more accurate components for intermolecular interactions.
- A new class, `OEFF::OEMMFFAmberOptions`, has been added that sets all options of the `OEFF::OEMMFFAmber` object during its construction.
- A new class, *OEMMFFIEFFOptions*, has been added that sets all options of the *OEMMFFIEFF* object during its construction.
- A new class, *OESheffieldOptions*, has been added that sets all options of the *OESheffield* object during its construction. All *OESheffield* methods with similar names as their *OESheffieldOptions* counterparts are now deprecated.

  The following table shows the older, deprecated functions and their new option replacements:

Deprecated	New option-based
OESheff::OESheffield::	*OESheffieldOptions.SetA*
OESheff::OESheffield::	*OESheffieldOptions.SetB*
OESheff::OESheffield::	*OESheffieldOptions.SetSoluteDielectricConstant*
OESheff::OESheffield::	*OESheffieldOptions.SetSolventDielectricConstant*
OESheff::OESheffield::(	*OESheffieldOptions.GetA OESheffieldOptions.GetB* *OESheffieldOptions.GetSoluteDielectricConstant* *OESheffieldOptions.GetSolventDielectricConstant*

CHAPTER

FIVE

COPYRIGHT AND TRADEMARKS

© 1997-2024 Cadence Design Systems, Inc. (Cadence), 2655 Seely Ave., San Jose, CA 95134, USA. All rights reserved.

OpenEye, Lexichem, ROCS, Grapheme, and Orion are registered trademarks of Cadence Design Systems, Inc.

This material contains proprietary information of Cadence Design Systems, Inc. Use of copyright notice is precautionary only and does not imply publication or disclosure.

The information supplied in this document is believed to be true, but no liability is assumed for its use or the infringement of the rights of others resulting from its use. Information in this document is subject to change without notice and does not represent a commitment on the part of Cadence Design Systems, Inc.

This package is sold/licensed/distributed subject to the condition that it shall not, by way of trade or otherwise, be lent, re-sold, hired out, or otherwise circulated without Cadence Design Systems' prior consent, in any form of packaging or cover other than that in which it was produced. No part of this manual or accompanying documentation, may be reproduced, stored in a retrieval system on optical or magnetic disk, tape, CD, DVD or other medium, or transmitted in any form or by any means, electronic, mechanical, photocopying recording, or otherwise for any purpose other than for the purchaser's personal use without a legal agreement or other written permission granted by Cadence.

This product should not be used in the planning, construction, maintenance, operation or use of any nuclear facility nor the flight, navigation, or communication of aircraft or ground support equipment. Cadence Design Systems, Inc., shall not be liable, in whole or in part, for any claims arising from such use, including death, bankruptcy or outbreak of war.

Windows is a registered trademark of Microsoft Corporation. Apple, OS X, macOS, and Macintosh are registered trademarks of Apple Inc. UNIX is a registered trademark of the Open Group. Linux is a registered trademark of Linus Torvalds. Red Hat is a registered trademark of Red Hat, Inc. SUSE, SLED, and SLES are registered trademarks of Novell, Inc. Ubuntu is a registered trademark of Canonical Ltd.

SYBYL is a registered trademark of Certara, L.P. MDL, BIOVIA and ISIS are registered trademarks of Dassault Systemes and/or its affiliates. SMARTS and SMIRKS are registered trademarks of Daylight Chemical Information Systems, Inc. MacroModel is a registered trademark of Schrodinger, LLC.

Python is a registered trademark of the Python Software Foundation. Django is a registered trademark of the Django Software Foundation. Java is a registered trademark of Oracle and/or its affiliates.

Other products and software packages referenced in this document are trademarks and registered trademarks of their respective vendors or manufacturers.

CHAPTER

SIX

SAMPLE CODE

For all of the code examples and samples within documents published by OpenEye, Cadence Molecular Sciences, the following terms of use apply.

TERMS FOR USE OF SAMPLE CODE

The software below ("Sample Code") is provided to current licensees or subscribers of Cadence products or SaaS offerings (each a "Customer"). Customer is hereby permitted to use, copy, and modify the Sample Code, subject to these terms. Cadence claims no rights to Customer's modifications. Modification of Sample Code is at Customer's sole and exclusive risk. Sample Code may require Customer to have a then current license or subscription to the applicable Cadence offering.

THE SAMPLE CODE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. CADENCE DISCLAIMS ALL WARRANTIES, INCLUDING, BUT NOT LIMITED TO, WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. In no event shall Cadence be liable for any damages or liability in connection with the Sample Code or its use.

CHAPTER

SEVEN

CITATION

Please follow the guidelines in this document to cite use of our products in research publications and other documents.

For more information, please contact OpenEye Support.

7.1 Orion®

To cite use of Orion, please use the following:

```
Orion <version-number>. OpenEye, Cadence Molecular Sciences, Santa Fe, NM. http://www.
↪eyesopen.com.
```

For example:

```
Orion 2025.1. OpenEye, Cadence Molecular Sciences, Santa Fe, NM. http://www.eyesopen.com.
```

The version number for Orion appears in the product's release notes, for example, https://docs.eyesopen.com/orion-user-interface/releasenotes/index.html.

7.2 Orion Floes

To cite use of an Orion-based floe package, please use the following:

```
OpenEye <package-name> <version-number>. OpenEye, Cadence Molecular Sciences, Santa Fe,␣
↪NM. http://www.eyesopen.com.
```

For example:

```
OpenEye Large-Scale Floes 4.2.3. OpenEye, Cadence Molecular Sciences, Santa Fe, NM.␣
↪http://www.eyesopen.com.
```

The version number for a floe package is displayed on the first page of the package's release notes. For example: https://docs.eyesopen.com/floe/modules/large_scale_floes/docs/source/releasenotes_2025_1.html.

7.3 Toolkits and Applications

To cite OpenEye toolkits or applications used in your work, please use the following:

```
OpenEye Toolkits [or Applications] <version-number>. OpenEye, Cadence Molecular Sciences,
↪ Santa Fe, NM. http://www.eyesopen.com.
```

For example:

```
OpenEye Toolkits 2024.2.1. OpenEye, Cadence Molecular Sciences, Santa Fe, NM. http://www.
↪eyesopen.com.
```

The Applications and Toolkits are released as a package, multiple times per year. The first part of the version number designates the calendar year. The version number appears on the summary document for the package.

If you want to cite an individual application or toolkit, you can use the syntax:

```
<product name> <version-number>. OpenEye, Cadence Molecular Sciences, Santa Fe, NM.␣
↪http://www.eyesopen.com.
```

where <*product name*> is:

- Bioisostere TK
- OEChem TK
- OEDepict TK
- OEDocking TK
- OEFF TK
- Eon TK
- FastROCS TK
- Grapheme TK
- GraphSim TK
- Lexichem TK
- OEMedChem TK
- MolProp TK
- Omega TK
- Quacpac TK
- Shape TK
- Spicoli TK
- Spruce TK
- Szmap TK
- Szybki TK
- Zap TK
- AFITT Application
- BROOD Application
- OEDocking Application
- EON Application
- OMEGA Application
- PICTO Application
- pKa-Prospector Application
- QUACPAC Application

- ROCS Application
- SiteHopper Application
- SPRUCE Application
- SZMAP Application
- SZYBKI Application
- VIDA Application

You can use the version number of the individual Application or Toolkit. For example:

```
SZYBKI Application 2.8.0.1. OpenEye, Cadence Molecular Sciences, Santa Fe, NM. http://
↪www.eyesopen.com.

Szybki TK 2.8.0.1. OpenEye, Cadence Molecular Sciences, Santa Fe, NM. http://www.
↪eyesopen.com.
```

The first documentation pages give the product names and version numbers, as in https://docs.eyesopen.com/applications/szybki/index.html and https://docs.eyesopen.com/toolkits/python/szybkitk/index.html.

7.4 Publications for Bibliographies

Please also cite these publications if the corresponding product was used in your research.

7.4.1 Orion

Sørensen, J., Bannan, C.C., Calabrò, G., Jain, V., Ovanesyan, G., Smith, A., Zhang, S., Bayly, C.I., Darden, T.A., Geballe, M.T., LeBard, D.N., McGann, M., Moon, J.B., Muddana, H.S., Shewmaker, A., LaFon, J., Tolbert, R.W., Skillman, A.G., Nicholls, A. Orion® A Cloud-Native Molecular Design Platform. In *Computational Drug Discovery*. V. Poongavanam and V. Ramaswamy, Eds. Wiley-VCH, 2024; pp. 579–615. DOI: 10.1002/9783527840748.ch24

7.4.2 AFITT and FLYNN Applications

Wlodek, S.; Skillman, A.G.; Nicholls, A. Automated Ligand Placement and Refinement with a Combined Force Field and Shape Potential. *Acta Crystallogr., Sect. D: Struct. Biol.* **2006**, *62* (7), 741–749. DOI: 10.1107/s0907444906016076

7.4.3 OEDocking Application and Toolkit

Kelley, B.P.; Brown, S.P.; Warren, G.L.; Muchmore, S.W. POSIT: Flexible Shape-Guided Docking For Pose Prediction. *J. Chem. Inf. Model.* **2015**, *55* (8), 1771-1780. DOI: 10.1021/acs.jcim.5b00142

McGann, M. FRED Pose Prediction and Virtual Screening Accuracy. *J. Chem. Inf. Model.* **2011**, *51* (3), 578-596. DOI: 10.1021/ci100436p

McGann, M. FRED and HYBRID Docking Performance on Standardized Datasets. *J. Comput.-Aided Mol. Des.* **2012**, *26* (8), 897-906. DOI: 10.1007/s10822-012-9584-8

7.4.4 OMEGA Application and Toolkit

If you are using OMEGA in torsion driving mode, please cite the following two publications.

Hawkins, P.C.D.; Skillman, A.G.; Warren, G.L.; Ellingson, B.A.; Stahl, M.T. Conformer Generation with OMEGA: Algorithm and Validation Using High Quality Structures from the Protein Databank and the Cambridge Structural Database. *J. Chem. Inf. Model.* **2010**, *50* (4), 572-584. DOI: 10.1021/ci100031x

Hawkins, P.C.D. and Nicholls, A. Conformer Generation with OMEGA: Learning from the Data Set and the Analysis of Failures. *J. Chem. Inf. Model.* **2012**, *52* (11), 2919–2936. DOI: /10.1021/ci300314k

If you are using OMEGA in macrocyle mode, please cite the following.

Hawkins, P.C.D. and Wlodek, S. Decisions with Confidence: Application to the Conformation Sampling of Molecules in the Solid State. *J. Chem. Inf. Model.* **2020**, *60* (7), 3518–3533. DOI: /10.1021/acs.jcim.0c00358

7.4.5 ROCS Application

Hawkins, P.C.D.; Skillman, A.G.; Nicholls, A. Comparison of Shape-Matching and Docking as Virtual Screening Tools. *J. Med. Chem.* **2007**, *50* (1), 74-82. DOI: 10.1021/jm0603365

7.5 OpenEye MMDS Web Service

To cite use of the Macromolecular Data Service (MMDS) web service, please use the syntax:

```
Macromolecular Data Service <version-number>. OpenEye, Cadence Molecular Sciences, Santa␣
↪Fe, NM. http://www.eyesopen.com.
```

For example:

```
Macromolecular Data Service 1.1. OpenEye, Cadence Molecular Sciences, Santa Fe, NM.␣
↪http://www.eyesopen.com.
```

The MMDS version number appears on the web service's release notes.

CHAPTER

EIGHT

TECHNOLOGY LICENSING

OpenEye products use the following technology under license.

Some of the open source technologies we use are licensed under the GNU Public License (GPL) version 2 or 3. In all instances, these technologies are communicated with using the "at arms length" principle, using pipes and command line arguments.

While the software is in some instances assembled into a container, that assembly does not invoke the GPL copyleft scope. For each Floe package where these are used, the environment log clearly details the installed programs and their version numbers. With the links below for each technology, the source for each of those programs is available.

Table 1: Licensed Technology

Name of Project	Website	License URL
_libgcc_mutex	https://github.com/conda-forge/_libgcc_mutex-feedstock	https://github.com/conda-forge/_libgcc_mutex-feedstock/blob/master/LICENSE.txt
_openmp_mutex	https://github.com/conda-forge/_openmp_mutex-feedstock	https://github.com/conda-forge/_openmp_mutex-feedstock/blob/main/LICENSE.txt
ablang2	https://github.com/TobiasHeOl/AbLang2	https://github.com/TobiasHeOl/AbLang2/blob/main/LICENSE
abseil-cpp	https://github.com/abseil/abseil-cpp	https://github.com/abseil/abseil-cpp/blob/master/LICENSE
absl-py	https://github.com/abseil/abseil-py	https://github.com/abseil/abseil-py/blob/main/LICENSE

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
aiohappyeyeballs	https://github.com/aio-libs/aiohappyeyeballs	https://github.com/aio-libs/aiohappyeyeballs/blob/main/LICENSE
aiohttp	https://docs.aiohttp.org/en/stable/	https://github.com/aio-libs/aiohttp/blob/master/LICENSE.txt
aiosignal	https://github.com/aio-libs/aiosignal	https://github.com/aio-libs/aiosignal/blob/master/LICENSE
Amazon Linux 2	https://aws.amazon.com/amazon-linux-2	N/A
AmberUtils	http://ambermd.org	N/A
ambit	https://github.com/khimaros/ambit	https://github.com/khimaros/ambit/blob/master/LICENSE
amqp	https://github.com/celery/py-amqp	https://github.com/celery/py-amqp/blob/master/LICENSE
anaconda-anon-usage	https://github.com/anaconda/anaconda-anon-usage	https://github.com/anaconda/anaconda-anon-usage/blob/main/LICENSE
angular	https://github.com/angular/angular.js	https://github.com/angular/angular.js/raw/master/LICENSE.md
angular-animate	https://github.com/angular/angular.js	https://github.com/angular/angular.js/raw/master/LICENSE.md
angular-cache	https://github.com/jmdobry/angular-cache	https://github.com/jmdobry/angular-cache/raw/master/LICENSE
angular-cookies	https://github.com/angular/angular.js	https://github.com/angular/angular.js/raw/master/LICENSE.md

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
angular-loggly-logger	https://github.com/ajbrown/angular-loggly-logger	https://github.com/ajbrown/angular-loggly-logger/raw/master/LICENSE.md
angular-mocks	https://github.com/angular/angular.js	https://raw.githubusercontent.com/angular/angular.js/master/LICENSE
angular-resource	https://github.com/angular/angular.js	https://raw.githubusercontent.com/angular/angular.js/master/LICENSE
angular-toggle-switch	https://github.com/cgarvis/angular-toggle-switch	https://github.com/cgarvis/angular-toggle-switch/blob/master/LICENSE
angular-ui-grid	https://github.com/angular-ui/ui-grid	https://github.com/angular-ui/ui-grid
angular-ui-router	https://github.com/angular-ui/ui-router	https://github.com/angular-ui/ui-router/raw/master/LICENSE
angular-ui-tree	https://github.com/angular-ui-tree/angular-ui-tree	https://github.com/angular-ui-tree/angular-ui-tree/raw/master/LICENSE
angular-vs-repeat	https://github.com/kamilkp/angular-vs-repeat	https://raw.githubusercontent.com/kamilkp/angular-vs-repeat/master/LICENSE
aniso8601	https://bitbucket.org/nielsenb/aniso8601	https://bitbucket.org/nielsenb/aniso8601/src/master/LICENSE
annotated-types	https://github.com/annotated-types/annotated-types	https://github.com/annotated-types/annotated-types/blob/main/LICENSE
antlr4-python3-runtime	http://www.antlr.org OR https://pypi.org/project/antlr4-python3-runtin	https://github.com/conda-forge/antlr4-python3-runtin/blob/master/LICENSE.txt

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
anyio	https://github.com/agronholm/anyio	https://github.com/agronholm/anyio/blob/master/LICENSE
appdirs	http://github.com/ActiveState/appdirs	http://github.com/ActiveState/appdirs/blob/master/LICENSE.txt
appengine	https://google.golang.org/appengine	https://github.com/golang/appengine/blob/master/LICENSE
arabic-reshaper	https://github.com/mpcabd/python-arabic-reshape	https://github.com/mpcabd/python-arabic-reshape/blob/master/LICENSE
archspec	https://github.com/archspec/archspec/blob/master/README.md	https://github.com/archspec/archspec/blob/master/COPYRIGHT
argon2-cffi	https://github.com/hynek/argon2-cffi	https://github.com/hynek/argon2-cffi/blob/master/LICENSE
argon2-cffi-bindings	https://github.com/hynek/argon2-cffi-bindings	https://github.com/hynek/argon2-cffi-bindings/blob/master/LICENSE
arpack	https://www.caam.rice.edu/software/ARPACK/	https://www.caam.rice.edu/software/ARPACK/
arrow	https://github.com/arrow-py/arrow	https://github.com/arrow-py/arrow?tab=Apache-2.0-1-ov-file#readme
ascii85	https://github.com/huandu/node-ascii85	https://github.com/huandu/node-ascii85/raw/master/LICENSE
ase	https://wiki.fysik.dtu.dk/ase/	https://wiki.fysik.dtu.dk/ase/development/licenseinfo.html#license-info
asgiref	https://github.com/django/asgiref/	https://github.com/django/asgiref/blob/main/LICENSE

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
asn1crypto	https://github.com/wbond/asn1crypto	https://github.com/wbond/asn1crypto/blob/master/LICENSE
assertions go-render	https://github.com/smartystreets/assertions/internal/go-render/render	https://github.com/smartystreets/assertions/blob/master/internal/go-render/LICENSE
assertions oglmatchers	https://github.com/smartystreets/assertions/internal/oglematchers	https://github.com/smartystreets/assertions/blob/master/internal/oglematchers/LICENSE
assertions	https://github.com/smartystreets/assertions	https://github.com/smartystreets/assertions/blob/master/LICENSE.md
asttokens	https://github.com/gristlabs/asttokens	https://github.com/gristlabs/asttokens/blob/master/LICENSE
astunparse	https://github.com/simonpercivall/astunparse	https://github.com/simonpercivall/astunparse/blob/master/LICENSE
async-lru	https://github.com/aio-libs/async-lru	https://github.com/aio-libs/async-lru/blob/master/LICENSE
async-timeout	https://github.com/aio-libs/async-timeout	https://github.com/aio-libs/async-timeout/blob/master/LICENSE
atk-1.0	https://docs.gtk.org/atk/	https://gitlab.gnome.org/GNOME/atk/-/blob/master/COPYING
atomic	https://github.com/uber-go/atomic	https://github.com/uber-go/atomic/blob/master/LICENSE.txt

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
atomicwrites	https://github.com/untitaker/python-atomicwrites	https://github.com/untitaker/python-atomicwrites/blob/master/LICENSE
attrs	https://www.attrs.org/	https://github.com/python-attrs/attrs/blob/main/LICENSE
aws-sdk-go	https://github.com/aws/aws-sdk-go	https://github.com/aws/aws-sdk-go/blob/master/LICENSE.txt
babel	https://babel.pocoo.org/ OR https://github.com/python-babel/babel	https://github.com/python-babel/babel/blob/master/LICENSE OR https://babel.pocoo.org/en/latest/license.html#babel-license
Babel	https://github.com/python-babel/babel	https://github.com/python-babel/babel/blob/master/LICENSE
backcall	https://github.com/takluyver/backcall	https://github.com/takluyver/backcall/blob/master/LICENSE
backports	https://github.com/brandon-rhodes/backports	https://github.com/brandon-rhodes/backports/blob/master/LICENSE.TXT
backports.functools-lru-cache	https://github.com/jaraco/backports.functools_lru_cache	https://github.com/jaraco/backports.functools_lru_cache/blob/main/LICENSE
backports.zstd	https://github.com/rogdham/backports.zstd	https://github.com/rogdham/backports.zstd?tab=License-1-ov-file#readme
base62	https://github.com/kare/base62	https://github.com/kare/base62/blob/main/LICENSE
beautifulsoup4	https://www.crummy.com/software/BeautifulSoup/	https://www.crummy.com/software/BeautifulSoup/

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
billiard	https://github.com/celery/billiard	https://github.com/celery/billiard/blob/master/LICENSE.txt
biopython	https://biopython.org	https://github.com/biopython/biopython/blob/master/LICENSE.rst
biotite	https://github.com/biotite-dev/biotite/blob/master/README.rst	https://github.com/biotite-dev/biotite/blob/master/LICENSE.rst
bitarray	https://github.com/ilanschnell/bitarray	https://github.com/ilanschnell/bitarray/blob/master/LICENSE
bitset	https://github.com/willf/bitset	https://github.com/willf/bitset/blob/master/LICENSE
blas	https://www.netlib.org/blas/	https://www.netlib.org/blas/#_licensing
bleach	https://github.com/mozilla/bleach	https://github.com/mozilla/bleach/blob/main/LICENSE
blessings	https://github.com/erikrose/blessings	https://github.com/erikrose/blessings/blob/master/LICENSE
blinker	https://pythonhosted.org/blinker/	https://github.com/pallets-eco/blinker/blob/master/LICENSE
bokeh	https://github.com/bokeh/bokeh	https://github.com/bokeh/bokeh/blob/branch-4.0/LICENSE.txt
blosc	https://github.com/Blosc/python-blosc	https://github.com/Blosc/python-blosc/blob/master/LICENSE.txt
blosc2	https://github.com/Blosc/python-blosc2	https://github.com/Blosc/c-blosc2/blob/main/LICENSE.txt

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
boltons	https://github.com/mahmoud/boltons	https://github.com/mahmoud/boltons/blob/master/LICENSE
Boltz	https://boltz.bio	https://github.com/jwohlwend/boltz/blob/main/LICENSE
boost	https://www.boost.org/doc/libs/1_71_0/libs/python/doc/html/index.html	https://www.boost.org/LICENSE_1_0.txt
boost-cpp	https://www.boost.org/doc/libs/1_71_0/libs/python/doc/html/index.html	https://www.boost.org/LICENSE_1_0.txt
bootstrap-vue	https://github.com/bootstrap-vue/bootstrap-vue	https://github.com/bootstrap-vue/bootstrap-vue/raw/master/LICENSE
boto3	https://github.com/boto/boto3	https://github.com/boto/boto3/blob/develop/LICENSE
botocore	https://github.com/boto/botocore	https://github.com/boto/botocore/blob/develop/LICENSE.txt
Bottleneck	https://bottleneck.readthedocs.io/en/latest/index.html	https://bottleneck.readthedocs.io/en/latest/license.html
Brotli	https://github.com/google/brotli	https://github.com/google/brotli/blob/master/LICENSE
brotli-bin	https://github.com/google/brotli	https://github.com/google/brotli/blob/master/LICENSE
brotli-python	http://python-hyper.org/projects/brotlipy/en/latest/	https://github.com/python-hyper/brotlicffi/blob/main/LICENSE
brotlicffi	https://github.com/python-hyper/brotlicffi	https://github.com/python-hyper/brotlicffi?tab=MIT-1-ov-file#readme
brotlipy	https://github.com/python-hyper/brotlicffi	https://github.com/python-hyper/brotlicffi/blob/main/LICENSE

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
bson	http://github.com/py-bson/bson	https://github.com/py-bson/bson/blob/master/LICENSE_APACHE
bytefmt	https://code.cloudfoundry.org/bytefmt	https://github.com/cloudfoundry/bytefmt/blob/master/LICENSE
bzip2	https://www.bzip.org	https://spdx.org/licenses/bzip2-1.0.6.html
c-ares	https://github.com/c-ares/c-ares	https://github.com/c-ares/c-ares/blob/main/LICENSE.md
ca-certificates	https://github.com/conda-forge/ca-certificates-feedsto	https://wiki.mozilla.org/CA/Included_Certificates
cached-property	https://github.com/pydanny/cached-property	https://github.com/pydanny/cached-property/blob/master/LICENSE
cachetools	https://github.com/tkem/cachetools/	https://github.com/tkem/cachetools/blob/master/LICENSE
cairo	https://pycairo.readthedocs.io/en/latest/	https://github.com/pygobject/pycairo/blob/master/COPYING
canvas	https://github.com/Automattic/node-canvas	https://github.com/Automattic/node-canvas
cctbx	https://github.com/cctbx/cctbx_project	https://github.com/cctbx/cctbx_project/blob/master/cctbx/LICENSE_2_0.txt
celery	https://github.com/celery/celery	https://github.com/celery/celery/blob/master/LICENSE
Cerberus	https://docs.python-cerberus.org/en/stable/	https://docs.python-cerberus.org/en/stable/license.html
certifi	https://certifiio.readthedocs.io/en/latest/	https://wiki.mozilla.org/index.php?title=CA/IncludedCertificates&redirect=no

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
cffi	https://github.com/python-cffi	https://github.com/cffi/cffi/blob/master/COPYRIGHT
cftime	https://pypi.org/project/cftime/	https://github.com/Unidata/cftime/blob/master/LICENSE
chardet	https://github.com/chardet/chardet	https://github.com/chardet/chardet/blob/master/LICENSE
charset-normalizer	https://github.com/ousret/charset_normalizer	https://github.com/Ousret/charset_normalizer/blob/master/LICENSE
ChEMBL Structure Pipeline	https://github.com/chembl/ChEMBL_Structure_Pipeline/tree/master	https://github.com/chembl/ChEMBL_Structure_Pipeline/blob/master/LICENSE
choreographer	https://github.com/mazpie/choreographer	https://github.com/mazpie/choreographer/blob/main/LICENSE
chunkreader	https://github.com/jackc/chunkreader/v2	https://github.com/jackc/chunkreader/blob/master/LICENSE
click	https://palletsprojects.com/p/click/	https://github.com/pallets/click/blob/main/LICENSE.rst
click-completion	https://github.com/click-contrib/click-completion	https://github.com/click-contrib/click-completion/blob/master/LICENSE
click-didyoumean	https://github.com/click-contrib/click-didyoumean	https://github.com/click-contrib/click-didyoumean/blob/master/LICENSE
click-plugins	https://github.com/click-contrib/click-plugins	https://github.com/click-contrib/click-plugins/blob/master/LICENSE.txt

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
click-repl	https://github.com/untitaker/click-repl	https://github.com/click-contrib/click-repl/blob/master/LICENSE
cloudpickle	https://github.com/cloudpipe/cloudpickle	https://github.com/cloudpipe/cloudpickle/blob/master/LICENSE
cmake	https://cmake.org/	https://cmake.org/licensing/
colorama	https://github.com/tartley/colorama	https://github.com/tartley/colorama/blob/master/LICENSE.txt
comm	https://github.com/ipython/comm	https://github.com/ipython/comm/blob/master/LICENSE
compose	https://github.com/docker/compose	https://github.com/docker/compose/blob/v2/LICENSE
conda-content-trust	https://github.com/conda/conda-content-trust	https://github.com/conda/conda-content-trust/blob/main/LICENSE
conda-libmamba-solver	https://github.com/conda/conda-libmamba-solv	https://github.com/conda/conda-libmamba-solv blob/main/LICENSE
conda-pack	https://conda.github.io/conda-pack/	https://github.com/conda/conda-pack/blob/main/LICENSE.txt
conda-package-handling	https://github.com/conda/conda-package-handli	https://github.com/conda/conda-package-handli blob/main/LICENSE
conda-package-streaming	https://anaconda.org/conda-forge/conda-package-stream	https://github.com/conda/conda-package-stream blob/main/LICENSE
conda-token	https://anaconda.org/anaconda/conda-token	https://anaconda.org/anaconda/conda-token

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
conda_package_strear	https://github.com/conda/conda-package-streamtree/main/conda_package_streaming	https://github.com/conda/conda-package-streamtab=License-1-ov-file
confuse	https://github.com/beetbox/confuse	https://github.com/beetbox/confuse/blob/main/LICENSE
contourpy	https://github.com/contourpy/contourpy	https://github.com/contourpy/contourpy/blob/main/LICENSE
cpp-expected	https://github.com/TartanLlama/expected	https://github.com/TartanLlama/expected?tab=CC0-1.0-1-ov-file#readme
cpp-peglib	https://github.com/yhirose/cpp-peglib	https://github.com/yhirose/cpp-peglib/blob/master/LICENSE
cryptography	https://github.com/pyca/cryptography	https://github.com/pyca/cryptography/blob/main/LICENSE
cssselect2	https://github.com/Kozea/cssselect2/	https://github.com/Kozea/cssselect2/blob/master/LICENSE
cuda-bindings	https://nvidia.github.io/cuda-python/cuda-bindings/latest/install.html	https://github.com/NVIDIA/cuda-python/blob/main/cuda_bindings/LICENSE
cuda-pathfinder	https://nvidia.github.io/cuda-python/cuda-pathfinder/latest/	https://nvidia.github.io/cuda-python/cuda-pathfinder/latest/license.html
cudatoolkit	https://developer.nvidia.com/cuda-toolkit	https://docs.nvidia.com/cuda/eula/index.html
cuequivariance-ops-cu12	https://developer.nvidia.com/cuequivariance	https://github.com/NVIDIA/cuEquivariance/blob/main/LICENSES/LICENSE

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
cuequivariance-ops-torch-cu12	https://developer.nvidia.com/cuequivariance	https://github.com/NVIDIA/cuEquivariance/blob/main/LICENSES/LICENSE
cuequivariance-torch	https://docs.nvidia.com/cuda/cuequivariance/	https://github.com/NVIDIA/cuEquivariance/blob/main/LICENSES/LICENSE
cupy-cuda113	https://cupy.dev/	https://github.com/cupy/cupy/blob/main/LICENSE
cupy-cuda11x	https://cupy.dev/	https://github.com/cupy/cupy/?tab=MIT-1-ov-file#readme
curl	https://curl.se	https://curl.se/docs/copyright.html
cycler	https://github.com/matplotlib/cycler	https://github.com/matplotlib/cycler/blob/main/LICENSE
cyrus-sasl	https://github.com/cyrusimap/cyrus-sasl	https://github.com/cyrusimap/cyrus-sasl/blob/master/COPYING
Cython	https://cython.org	https://www.apache.org/licenses/LICENSE-2.0.html
d3	https://github.com/mbostock/d3	https://github.com/mbostock/d3/raw/master/LICENSE
dataclasses	https://github.com/ericvsmith/dataclasses	https://github.com/ericvsmith/dataclasses/blob/master/LICENSE.txt
ddsketch	http://github.com/datadog/sketches-py	https://github.com/DataDog/sketches-py/blob/master/LICENSE
debugpy	https://aka.ms/debugpy	https://github.com/microsoft/debugpy/blob/main/LICENSE

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
decimal	https://github.com/shopspring/decimal	https://github.com/shopspring/decimal/blob/master/LICENSE
decorator	https://github.com/micheles/decorator	https://github.com/micheles/decorator/blob/master/LICENSE.txt
deepdiff	https://github.com/seperman/deepdiff	https://github.com/seperman/deepdiff/blob/master/LICENSE
deeptime	https://github.com/deeptime-ml/deeptime	https://github.com/deeptime-ml/deeptime/blob/main/LICENSE.txt
defusedxml	https://github.com/tiran/defusedxml	https://github.com/tiran/defusedxml/blob/main/LICENSE
dftd3	https://github.com/dftd3/tad-dftd3	https://github.com/dftd3/tad-dftd3/blob/main/LICENSE
dill	https://github.com/uqfoundation/dill	https://github.com/uqfoundation/dill/blob/master/LICENSE
distlib	https://distlib.readthedocs.io/en/latest/	https://github.com/pypa/distlib/blob/master/LICENSE.txt
distro	https://github.com/python-distro/distro	https://github.com/python-distro/distro/blob/master/LICENSE
Django	https://www.djangoproject.com/	https://github.com/django/django/blob/main/LICENSE
django-classy-tags	https://github.com/django-cms/django-classy-tags	https://github.com/django-cms/django-classy-tags/blob/master/LICENSE
django-cors-headers	https://github.com/adamchainz/django-cors-headers	https://github.com/adamchainz/django-cors-headers/blob/main/LICENSE

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
django-csp	https://github.com/mozilla/django-csp	https://github.com/mozilla/django-csp/blob/main/LICENSE
django-extensions	https://github.com/django-extensions/django-extensions	https://github.com/django-extensions/django-extensions/blob/main/LICENSE
django-filter	https://github.com/carltongibson/django-filter/tree/master	https://github.com/carltongibson/django-filter/blob/main/LICENSE
django-redis	https://github.com/jazzband/django-redis	https://github.com/jazzband/django-redis/blob/master/LICENSE
django-taggit	https://github.com/jazzband/django-taggit	https://github.com/jazzband/django-taggit/blob/master/LICENSE
django-taggit-serializer	https://github.com/glemmaPaul/django-taggit-serializ	https://github.com/glemmaPaul/django-taggit-serializ blob/master/LICENSE
django-taggit-templatetags2	https://github.com/fizista/django-taggit-templat	https://github.com/fizista/django-taggit-templat blob/master/LICENSE.txt
djangorestframework	https://www.django-rest-framewor org/	https://github.com/encode/django-rest-framewor blob/master/LICENSE.md
dkh	https://psicode.org/psi4manual/master/dkh.html	https://github.com/psi4/dkh/blob/master/COPYING.LESSER
dm-tree	https://github.com/deepmind/tree	https://github.com/deepmind/tree/blob/master/LICENSE
docker-py	https://github.com/docker/docker-py/	https://github.com/docker/docker-py/blob/master/LICENSE

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
docker-pycreds	https://github.com/shin-/dockerpy-creds	https://github.com/shin-/dockerpy-creds/blob/master/LICENSE
docopt	https://docopt.org	https://github.com/docopt/docopt/blob/master/LICENSE-MIT
docutils	https://docutils.sourceforge.io	https://docutils.sourceforge.io/COPYING.html
drf-dynamic-fields	https://github.com/dbrgn/drf-dynamic-fields	https://github.com/dbrgn/drf-dynamic-fields/blob/master/LICENSE
DuckDB	https://duckdb.org/	https://github.com/duckdb/duckdb?tab=MIT-1-ov-file#readme
edgembar	https://rutgerslbsr.gitlab.io/fe-toolkit/edgembar/index.html	https://gitlab.com/RutgersLBSR/fe-toolkit/-/blob/master/LICENSE
editdistance	https://github.com/roy-ht/editdistance	https://github.com/roy-ht/editdistance/blob/master/LICENSE
eigen	https://eigen.tuxfamily.org/index.php?title=Main_Page	https://eigen.tuxfamily.org/index.php?title=Main_Page#License
einops	https://github.com/arogozhnikov/einops	https://github.com/arogozhnikov/einops/blob/master/LICENSE
einx	https://github.com/fferflo/einx	https://github.com/fferflo/einx/blob/master/LICENSE
entrypoints	https://github.com/takluyver/entrypoints	https://github.com/takluyver/entrypoints/blob/master/LICENSE
errors	https://github.com/pkg/errors	https://github.com/pkg/errors/blob/master/LICENSE

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
eslint-plugin	https://github.com/typescript-eslint/typescript-eslint	https://github.com/typescript-eslint/typescript-eslint/raw/master/LICENSE
et_xmlfile	https://foss.heptapod.net/openpyxl/et_xmlfile	https://foss.heptapod.net/openpyxl/et_xmlfile/-/blob/branch/default/LICENCE.rst
exceptiongroup	https://github.com/agronholm/exceptiongroup	https://github.com/agronholm/exceptiongroup/blob/main/LICENSE
execnet	https://execnet.readthedocs.io/en/latest/	https://github.com/pytest-dev/execnet/blob/master/LICENSE
executing	https://github.com/alexmojaki/executing	https://github.com/alexmojaki/executing/blob/master/LICENSE.txt
expat	https://libexpat.github.io	https://github.com/libexpat/libexpat/blob/master/expat/COPYING
fairscale	https://github.com/facebookresearch/fairscale	https://github.com/facebookresearch/fairscale/blob/main/LICENSE
fairseq	https://github.com/pytorch/fairseq	https://github.com/facebookresearch/fairseq/blob/main/LICENSE
fastjsonschema	https://github.com/horejsek/python-fastjsonschem	https://github.com/horejsek/python-fastjsonschem/blob/master/LICENSE
fastrlock	https://github.com/scoder/fastrlock	https://github.com/scoder/fastrlock/blob/master/LICENSE
fftw	https://www.fftw.org	commercial license

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
filebuffer	https://github.com/mattetti/filebuffer	https://github.com/mattetti/filebuffer/blob/master/LICENSE
filelock	https://py-filelock.readthedocs.io/en/latest/index.html	https://github.com/tox-dev/py-filelock/blob/main/LICENSE
finufft	https://github.com/flatironinstitute/finufft	https://github.com/flatironinstitute/finufft/blob/master/LICENSE
Flask	https://flask.palletsprojects.com/en/2.1.x/	https://flask.palletsprojects.com/en/2.1.x/license/
flatbuffers	https://google.github.io/flatbuffers/	https://github.com/google/flatbuffers/blob/master/LICENSE.txt
flexcache	https://github.com/hgrecco/flexcache	https://github.com/hgrecco/flexcache/blob/main/LICENSE
flexparser	https://github.com/hgrecco/flexparser	https://github.com/hgrecco/flexparser/blob/main/LICENSE
flit-core	https://github.com/pypa/flit	https://github.com/scientific-python/lazy_loader/blob/main/LICENSE.md
FLTK	https://www.fltk.org/index.php	https://www.fltk.org/COPYING.php
fmt	https://fmt.dev/latest/index.html	https://github.com/fmtlib/fmt/blob/master/LICENSE.rst
font-awesome	https://github.com/FortAwesome/Font-Awesome	https://github.com/FortAwesome/Font-Awesome
font-ttf-dejavu-sans-mono	https://dejavu-fonts.github.io	https://dejavu-fonts.github.io/License.html
font-ttf-inconsolata	https://fonts.google.com/specimen/Inconsolata	https://scripts.sil.org/cms/scripts/page.php?site_id=nrsi&id=OFL

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
font-ttf-source-code-pro	https://fonts.google.com/specimen/Source+Code+Pro	https://scripts.sil.org/cms/scripts/page.php?site_id=nrsi&id=OFL
font-ttf-ubuntu	https://fonts.google.com/specimen/Ubuntu	https://ubuntu.com/legal/font-licence
fontconfig	https://www.freedesktop.org/wiki/Software/fontconfig/	https://gitlab.freedesktop.org/fontconfig/fontconfig/-/blob/main/COPYING
fonts-conda-ecosystem	https://anaconda.org/conda-forge/fonts-conda-ecosyster	https://docs.conda.io/en/latest/license.html
fonts-conda-forge	https://anaconda.org/conda-forge/fonts-conda-forge/	https://creativecommons.org/licenses/by/4.0/
fonttools	https://github.com/fonttools/fonttools	https://github.com/fonttools/fonttools/blob/main/LICENSE
forcebalance	https://simtk.org/home/forcebalance OR https://github.com/leeping/forcebalance	https://github.com/leeping/forcebalance/blob/master/LICENSE
fqdn	https://github.com/ypcrts/fqdn	https://github.com/ypcrts/fqdn?tab=MPL-2.0-1-ov-file#readme
freetype	https://freetype.org	https://freetype.org/license.html
freetype-py	https://github.com/rougier/freetype-py	https://github.com/rougier/freetype-py?tab=License-1-ov-file#readme
fribidi	https://github.com/fribidi/fribidi	https://github.com/fribidi/fribidi/blob/master/COPYING
frozendict	https://github.com/Marco-Sulla/python-frozendict	https://github.com/Marco-Sulla/python-frozendict/blob/master/LICENSE.txt
frozenlist	https://github.com/aio-libs/frozenlist	https://github.com/aio-libs/frozenlist/blob/master/LICENSE

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
fsmlite	https://github.com/tkem/fsmlite	https://github.com/tkem/fsmlite/blob/master/LICENSE
fsspec	https://github.com/fsspec/filesystem_spec	https://github.com/fsspec/filesystem_spec/blob/master/LICENSE
funcy	https://github.com/Suor/funcy	https://github.com/Suor/funcy/blob/master/LICENSE
gast	https://github.com/serge-sans-paille/gast/	https://github.com/serge-sans-paille/gast/blob/master/LICENSE
gau2grid	https://github.com/dgasmith/gau2grid	https://github.com/dgasmith/gau2grid/blob/master/LICENSE
gax-go	https://github.com/googleapis/gax-go/v2	https://github.com/googleapis/gax-go/blob/main/LICENSE
gdk-pixbuf	https://gitlab.gnome.org/GNOME/gdk-pixbuf	https://gitlab.gnome.org/GNOME/gdk-pixbuf/-/blob/master/COPYING
gemmi	https://gemmi.readthedocs.io/en/latest/	https://github.com/project-gemmi/gemmi/blob/master/LICENSE.txt
genproto	https://google.golang.org/genproto/googleapis	https://github.com/googleapis/go-genproto/blob/main/LICENSE
geometric	https://openbase.com/python/geometric	https://openbase.com/python/geometric
giflib	https://giflib.sourceforge.net	https://github.com/mirrorer/giflib/blob/master/COPYING
gitdb	https://github.com/gitpython-developers/gitdb	https://github.com/gitpython-developers/gitdb/blob/master/LICENSE
GitPython	https://github.com/gitpython-developers/GitPython	https://github.com/gitpython-developers/GitPython/blob/main/LICENSE

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
glib	https://docs.gtk.org/glib/	https://gitlab.gnome.org/GNOME/glib/-/blob/main/LICENSES/LGPL-2.1-or-later.txt
GLM Library	https://github.com/g-truc/glm	https://github.com/g-truc/glm/blob/master/manual.md#section0
gls	https://github.com/jtolds/gls	https://github.com/jtolds/gls/blob/master/LICENSE
gmpy2	https://github.com/aleaxit/gmpy	https://github.com/gmpy2/gmpy2/blob/master/COPYING.LESSER
go-cleanhttp	https://github.com/hashicorp/go-cleanhttp	https://github.com/hashicorp/go-cleanhttp/blob/master/LICENSE
go-connections	https://github.com/docker/go-connections	https://github.com/docker/go-connections/blob/master/LICENSE
go-cpio	https://github.com/cavaliercoder/go-cpio	https://github.com/cavaliercoder/go-cpio/blob/master/LICENSE
go-getter	https://github.com/hashicorp/go-getter	https://github.com/hashicorp/go-getter/blob/master/LICENSE
go-homedir	https://github.com/mitchellh/go-homedir	https://github.com/mitchellh/go-homedir/blob/master/LICENSE
go-ini	https://github.com/go-ini/ini	https://github.com/go-ini/ini/blob/master/LICENSE
go-jmespath	https://github.com/jmespath/go-jmespath	https://github.com/jmespath/go-jmespath/blob/master/LICENSE

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
go-junit-report	https://github.com/jstemmer/go-junit-report	https://github.com/jstemmer/go-junit-report/blob/master/LICENSE
go-netrc	https://github.com/bgentry/go-netrc/netrc	https://github.com/bgentry/go-netrc/blob/master/LICENSE
go-ole	https://github.com/go-ole/go-ole	https://github.com/go-ole/go-ole/blob/master/LICENSE
go-pg	https://github.com/go-pg/pg	https://github.com/go-pg/pg/blob/master/LICENSE
go-redis	https://github.com/go-redis/redis/v8	https://github.com/go-redis/redis/blob/master/LICENSE
go-rendezvous	https://github.com/dgryski/go-rendezvous	https://github.com/dgryski/go-rendezvous/blob/master/LICENSE
go-safetemp	https://github.com/hashicorp/go-safetemp	https://github.com/hashicorp/go-safetemp/blob/master/LICENSE
go-sysconf	https://github.com/tklauser/go-sysconf	https://github.com/tklauser/go-sysconf/blob/master/LICENSE
go-testing-interface	https://github.com/mitchellh/go-testing-interface	https://github.com/mitchellh/go-testing-interface/blob/master/LICENSE
go-units	https://github.com/docker/go-units	https://github.com/docker/go-units/blob/master/LICENSE
go-version	https://github.com/hashicorp/go-version	https://github.com/hashicorp/go-version/blob/master/LICENSE
go-zglob	https://github.com/mattn/go-zglob	https://github.com/mattn/go-zglob/blob/master/LICENSE

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
go.opencensus	https://go.opencensus.io	https://github.com/census-instrumentatio opencensus-go/blob/master/LICENSE
gobrake.v2	https://gopkg.in/airbrake/gobrake.v2	https://github.com/airbrake/gobrake/blob/v2.0.9/LICENSE
goconvey	https://github.com/smartystreets/goconvey	https://github.com/smartystreets/goconvey/blob/master/LICENSE.md
golden-layout	https://github.com/deepstreamIO/golden-layout	https://github.com/deepstreamIO/golden-layout/raw/master/LICENSE
google-auth	https://google-auth.readthedocs.io/en/master/	https://github.com/GoogleCloudPlatform google-auth-library-p blob/master/LICENSE
google-auth-oauthlib	https://github.com/googleapis/google-auth-library-p	https://github.com/googleapis/google-auth-library-p blob/main/LICENSE
google-cloud-go	https://cloud.google.com/go	https://github.com/googleapis/google-cloud-go/blob/main/LICENSE
google-cloud-go/storage	https://cloud.google.com/go/storage	https://github.com/googleapis/google-cloud-go/blob/main/LICENSE
google-pasta	https://github.com/google/pasta	https://github.com/google/pasta/blob/master/LICENSE
google.golang.org/api	https://google.golang.org/api	https://github.com/googleapis/google-api-go-client/blob/main/LICENSE
gopsutil	https://github.com/shirou/gopsutil	https://github.com/shirou/gopsutil/blob/master/LICENSE

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
gorilla websocket	https://github.com/gorilla/websocket	https://github.com/gorilla/websocket/blob/master/LICENSE
gpytorch	https://gpytorch.ai	https://github.com/cornellius-gp/gpytorch?tab=MIT-1-ov-file#readme
graphite2	https://github.com/silnrsi/graphite	https://github.com/silnrsi/graphite/blob/master/LICENSE
graphviz	https://graphviz.org/	https://graphviz.org/license/
greenlet	https://greenlet.readthedocs.io/en/latest/	https://github.com/python-greenlet/greenlet/blob/master/LICENSE
groupcache	https://github.com/golang/groupcache	https://github.com/golang/groupcache/blob/master/LICENSE
grpc	https://google.golang.org/grpc	https://github.com/grpc/grpc-go/blob/master/LICENSE
grpc-cpp	https://github.com/grpc/grpc	https://github.com/grpc/grpc/blob/master/LICENSE
grpcio	https://github.com/grpc/grpc.io/blob/main/LICENSE	https://github.com/grpc/grpc.io/blob/main/LICENSE
gtk2	https://gitlab.gnome.org/GNOME/gtk	https://gitlab.gnome.org/GNOME/gtk/-/blob/main/COPYING
gts	https://sourceforge.net/projects/gts/	https://www.gnu.org/licenses/lgpl-3.0.html
h11	https://github.com/python-hyper/h11	https://github.com/python-hyper/h11/blob/master/LICENSE.txt
h2	https://github.com/python-hyper/h2	https://github.com/python-hyper/h2?tab=MIT-1-ov-file#readme

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
h5py	https://www.h5py.org	https://docs.h5py.org/en/stable/licenses.html
harfbuzz	https://github.com/harfbuzz/harfbuzz	https://github.com/harfbuzz/harfbuzz/blob/main/COPYING
hdbscan	https://hdbscan.readthedocs.io/en/latest/	https://github.com/scikit-learn-contrib/hdbscan/blob/master/LICENSE
hdf4	https://www.hdfgroup.org/solutions/hdf4/	https://github.com/HDFGroup/hdf4/blob/master/COPYING
hdf5	https://www.hdfgroup.org/solutions/hdf5/	https://github.com/HDFGroup/hdf5/blob/master/COPYING
he	https://github.com/mathiasbynens/he	https://github.com/mathiasbynens/he/raw/master/LICENSE-MIT.txt
hpack	https://github.com/python-hyper/hpack	https://github.com/python-hyper/hpack?tab=MIT-1-ov-file#readme
html-loader	https://github.com/webpack-contrib/html-loader	https://github.com/webpack-contrib/html-loader/raw/master/LICENSE
html5lib	https://github.com/html5lib/html5lib-python	https://github.com/html5lib/html5lib-python/blob/master/LICENSE
htslib	https://www.htslib.org	https://github.com/samtools/htslib/blob/develop/LICENSE
httpcore	https://www.encode.io/httpcore/	https://github.com/encode/httpcore/blob/master/LICENSE.md
httpx	https://github.com/encode/httpx	https://github.com/encode/httpx/blob/master/LICENSE.md

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
humanize	https://github.com/jmoiron/humanize	https://github.com/jmoiron/humanize/blob/master/LICENCE
hydra-core	https://github.com/facebookresearch/hydra	https://github.com/facebookresearch/hydra/blob/main/LICENSE
hyperframe	https://github.com/python-hyper/hyperframe	https://github.com/python-hyper/hyperframe?tab=MIT-1-ov-file#readme
icu	https://github.com/unicode-org/icu	https://github.com/unicode-org/icu/blob/main/icu4c/LICENSE
idna	https://github.com/kjd/idna	https://github.com/kjd/idna/blob/master/LICENSE.md
ihm	https://github.com/ihmwg/python-ihm	https://github.com/ihmwg/python-ihm/blob/main/LICENSE
imageio	https://github.com/imageio/imageio	https://github.com/imageio/imageio/blob/master/LICENSE
imagesize	https://github.com/shibukawa/imagesize_py	https://github.com/shibukawa/imagesize_py/blob/master/LICENSE.rst
ImmuneBuilder	https://github.com/oxpig/ImmuneBuilder/tree/main	https://github.com/oxpig/ImmuneBuilder/blob/main/LICENSE
importlib-metadata	https://github.com/python/importlib_metadata	https://github.com/python/importlib_metadata/blob/main/LICENSE
importlib_metadata	https://github.com/python/importlib_metadata	https://github.com/python/importlib_metadata/blob/main/LICENSE

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
importlib_resources	https://github.com/python/importlib_resources	https://github.com/python/importlib_resources/blob/main/LICENSE
InChI	https://iupac.org/who-we-are/divisions/division-details/inchi/	https://iupac.org/wp-content/uploads/2021/06/iupac-inchi-license_2020.pdf
inflection	https://github.com/jinzhu/inflection	https://github.com/jinzhu/inflection/blob/master/LICENSE
ini.v1	https://gopkg.in/ini.v1	https://github.com/go-ini/ini/blob/v1.66.2/LICENSE
iniconfig	https://github.com/pytest-dev/iniconfig	https://github.com/pytest-dev/iniconfig/blob/main/LICENSE
innersvg-polyfill	https://github.com/dnozay/innersvg-polyfill	https://github.com/dnozay/innersvg-polyfill/raw/master/LICENSE
intel-openmp	https://github.com/hermitcore/libomp_oss	https://github.com/hermitcore/libomp_oss/blob/hermit/LICENSE.txt
ipykernel	https://ipython.org	https://ipython.org
ipython	https://ipython.org	https://github.com/ipython/ipython/blob/master/LICENSE
ipython-genutils	http://ipython.org	https://github.com/ipython/ipython_genutils/blob/master/COPYING.md
ipython_pygments_le	https://github.com/ipython/ipython-pygments-lex	https://github.com/ipython/ipython-pygments-lex blob/main/LICENSE
ipywidgets	http://jupyter.org	https://github.com/jupyter-widgets/ipywidgets/blob/main/LICENSE

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
isodate	https://github.com/gweis/isodate/	https://github.com/gweis/isodate/blob/master/LICENSE
isoduration	https://github.com/bolsote/isoduration	https://github.com/bolsote/isoduration?tab=ISC-1-ov-file#readme
itsdangerous	https://palletsprojects.com/p/itsdangerous/	https://itsdangerous.palletsprojects.com/en/2.1.x/license/
jax	https://github.com/google/jax	https://github.com/google/jax/blob/main/LICENSE
jaxlib	https://github.com/google/jax	https://github.com/google/jax/blob/main/LICENSE
jaxtyping	https://github.com/patrick-kidger/jaxtyping	https://github.com/patrick-kidger/jaxtyping?tab=License-1-ov-file#readme
jedi	https://jedi.readthedocs.io/en/latest/	https://github.com/davidhalter/jedi/blob/master/LICENSE.txt
Jinja2	https://palletsprojects.com/p/jinja/	https://github.com/pallets/jinja/blob/main/LICENSE.rst
jmespath	https://github.com/jmespath/jmespath.py	https://github.com/jmespath/jmespath.py/blob/develop/LICENSE.txt
joblib	https://joblib.readthedocs.io/en/latest/	https://github.com/joblib/joblib/blob/master/LICENSE.txt
jpeg	https://www.ijg.org	https://www.ijg.org/files/README
json5	https://github.com/dpranke/pyjson5	https://github.com/dpranke/pyjson5/blob/main/LICENSE
jsonfield	https://github.com/dmkoch/django-jsonfield/	https://github.com/rpkilby/jsonfield/blob/master/LICENSE

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
jsonpatch	https://github.com/stefankoegl/python-json-patch	https://github.com/stefankoegl/python-json-patch/blob/master/LICENSE
jsonpickle	https://github.com/jsonpickle/jsonpickle	https://github.com/jsonpickle/jsonpickle/blob/main/LICENSE
jsonpointer	https://github.com/stefankoegl/python-json-pointer	https://github.com/stefankoegl/python-json-pointer/blob/master/LICENSE.txt
jsonschema	https://github.com/python-jsonschema/jsonschema	https://github.com/python-jsonschema/jsonschema/blob/master/COPYING
jsonschema-specifications	https://github.com/python-jsonschema/jsonschema-specificat	https://github.com/python-jsonschema/jsonschema-specificat blob/main/COPYING
jstat	https://github.com/jstat/jstat	https://www.opensource.org/licenses/mit-license.php
jupyter-client	https://jupyter.org	https://github.com/jupyter/jupyter_client/blob/main/COPYING.md
jupyter-core	https://jupyter.org	https://github.com/jupyter/jupyter_core/blob/master/COPYING.md
jupyter-events	https://github.com/jupyter/jupyter_events	https://github.com/jupyter/jupyter_events/blob/main/LICENSE
jupyter-lsp	https://github.com/jupyter-lsp/jupyterlab-lsp	https://github.com/jupyter-lsp/jupyterlab-lsp
jupyter-serverhttp://jupyter.or	https://github.com/jupyter-server/jupyter_server/blob/main/LICENSE	
jupyter_client	http://jupyter.org	https://github.com/jupyter/jupyter_client/blob/main/LICENSE

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
jupyter_core	http://jupyter.org	https://github.com/jupyter/jupyter_core/blob/master/LICENSE
jupyter_server	https://github.com/jupyter-server/jupyter_server	https://github.com/jupyter-server/jupyter_server/blob/main/LICENSE
jupyter_server_termin	https://github.com/jupyter-server/jupyter_server_terminals	https://github.com/jupyter-server/jupyter_server_terminals/blob/main/LICENSE
jupyterlab	https://github.com/jupyterlab/jupyterlab	https://github.com/jupyterlab/jupyterlab/blob/main/LICENSE
jupyterlab-pygments	http://jupyter.org	https://github.com/jupyterlab/jupyterlab_pygments/blob/main/LICENSE
jupyterlab-widgets	http://jupyter.org	https://github.com/jupyter-widgets/ipywidgets/blob/master/LICENSE
jupyter-lab_pygments	https://github.com/jupyterlab/jupyterlab_pygments	https://github.com/jupyterlab/jupyterlab_pygments?tab=BSD-3-Clause-1-ov-file#readme
jupyterlab_server	https://github.com/jupyterlab/jupyterlab_server	https://github.com/jupyterlab/jupyterlab_server/blob/main/LICENSE
jupyterlab_widgets	https://github.com/jupyter-widgets/ipywidgets	https://github.com/jupyter-widgets/ipywidgets?tab=BSD-3-Clause-1-ov-file#readme
kaleido	https://github.com/plotly/Kaleido	https://github.com/plotly/Kaleido/blob/master/LICENSE.txt
keras	https://github.com/keras-team/keras	https://github.com/keras-team/keras/blob/master/LICENSE

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
Keras-Preprocessing	https://github.com/keras-team/keras-preprocessing	https://github.com/keras-team/keras-preprocessing/blob/master/LICENSE
keras-tuner	https://github.com/keras-team/keras-tuner	https://github.com/keras-team/keras-tuner/blob/master/LICENSE
keyring	https://github.com/jaraco/keyring	https://github.com/jaraco/keyring/blob/main/LICENSE
keyutils	https://github.com/sassoftware/python-keyutils	https://github.com/sassoftware/python-keyutils/blob/master/LICENSE
kiwisolver	https://kiwisolver.readthedocs.io/en/latest/	https://github.com/nucleic/kiwi/blob/main/LICENSE
kombu	https://kombu.readthedocs.io	https://github.com/celery/kombu/blob/master/LICENSE
krb5	https://web.mit.edu/kerberos/	https://github.com/krb5/krb5/blob/master/src/lib/gssapi/LICENSE
kt-legacy	https://github.com/haifeng-jin/kt-legacy	https://github.com/haifeng-jin/kt-legacy/blob/main/LICENSE
lark	https://github.com/lark-parser/lark	https://github.com/lark-parser/lark?tab=MIT-1-ov-file#readme
lazy_loader	https://github.com/scientific-python/lazy_loader	https://github.com/scientific-python/lazy_loader/blob/main/LICENSE.md
lcms2	https://www.littlecms.com	https://github.com/sk1project/python-lcms2/blob/master/LICENSE
legacy-cgi	https://github.com/jackrosenthal/legacy-cgi	https://github.com/jackrosenthal/legacy-cgi?tab=License-1-ov-file#readme

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
lerc	https://github.com/Esri/lerc	https://github.com/Esri/lerc/blob/master/LICENSE
libarchive	http://www.libarchive.org	https://raw.githubusercontent.com/libarchive/libarchive/master/COPYING
libblas	https://www.netlib.org/blas/	https://www.netlib.org/blas/#_licensing
libbrotlicommon	https://github.com/google/brotli	https://github.com/google/brotli/blob/master/LICENSE
libbrotlidec	https://github.com/google/brotli	https://github.com/google/brotli/blob/master/LICENSE
libbrotlienc	https://github.com/google/brotli	https://github.com/google/brotli/blob/master/LICENSE
libcblas	https://anaconda.org/conda-forge/libcblas	N/A
libclang	https://github.com/sighingnow/libclang	https://github.com/sighingnow/libclang/blob/master/LICENSE.TXT
libcurl	https://curl.se/libcurl/	https://curl.se/docs/copyright.html
libcxx	https://github.com/llvm-mirror/libcxx	https://github.com/llvm-mirror/libcxx/blob/master/LICENSE.TXT
libdb	https://www.oracle.com/technology/software/products/berkeley-db/index.html	https://github.com/berkeleydb/libdb/blob/master/LEGAL
libdeflate	https://github.com/ebiggers/libdeflate	https://github.com/ebiggers/libdeflate/blob/master/COPYING
libedit	https://thrysoee.dk/editline/	http://www.netbsd.org/Goals/redistribution.html
libev	https://software.schmorp.de/pkg/libev.html	https://software.schmorp.de/pkg/libev.html

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
libexpat	https://libexpat.github.io/	https://github.com/libexpat/libexpat?tab=MIT-1-ov-file#readme
libffi	https://github.com/libffi/libffi	https://github.com/libffi/libffi/blob/master/LICENSE
libgcc	https://gcc.gnu.org/onlinedocs/gccint/Libgcc.html	https://gcc.gnu.org/onlinedocs/libstdc++/manual/license.html
libgcrypt	https://gnupg.org/software/index.html	https://git.gnupg.org/cgi-bin/gitweb.cgi?p=libgcrypt.git;a=blob;f=COPYING.LIB;hb=HEAD
libgd	https://libgd.github.io	https://libgd.github.io/manuals/2.2.1/files/license-txt.html
libglib	https://github.com/PolMine/libglib	https://github.com/PolMine/libglib/blob/master/include/glib-2.0/glib.h
libiconv	https://www.gnu.org/software/libiconv/	https://www.gnu.org/software/libiconv/#downloading
libint	https://tinyurl.com/yvw97wbw	https://tinyurl.com/yvw97wbw
liblapack	http://www.netlib.org/lapack/	https://www.netlib.org/lapack/LICENSE.txt
liblapacke	https://anaconda.org/conda-forge/liblapacke	https://www.netlib.org/lapack/LICENSE.txt
liblzma	https://tukaani.org/xz/	https://git.tukaani.org/?p=xz.git;a=blob;f=COPYING.0BSD
libmamba	https://medium.com/@QuantStack/open-software-packag	https://github.com/mamba-org/mamba/blob/main/LICENSE

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
libmambapy	https://www.anaconda.com/blog/a-faster-conda-for-a-g	https://github.com/conda/conda-libmamba-solv blob/main/LICENSE
libnetcdf	https://www.unidata.ucar.edu/software/netcdf/	https://www.unidata.ucar.edu/software/netcdf/copyright.html
libnghttp2	https://github.com/nghttp2/nghttp2	https://github.com/nghttp2/nghttp2/blob/master/COPYING
libnsl	https://github.com/thkukuk/libnsl	https://gcc.gnu.org/onlinedocs/libstdc++/manual/license.html
libopenblas	https://www.openblas.net/	https://github.com/xianyi/OpenBLAS/blob/develop/LICENSE
libpng	https://www.libpng.org/pub/png/libpng.html	https://www.libpng.org/pub/png/src/libpng-LICENSE.txt
libpq	https://www.postgresql.org/	https://www.postgresql.org/about/licence/
librsvg	https://gitlab.gnome.org/GNOME/librsvg	https://gitlab.gnome.org/GNOME/librsvg/-/blob/main/COPYING.LIB
libsolv	https://github.com/openSUSE/libsolv	https://github.com/openSUSE/libsolv/blob/master/LICENSE.BSD
libsqlite	https://github.com/mackyle/sqlite	https://github.com/mackyle/sqlite?tab=License-1-ov-file#readme
libssh2	https://github.com/libssh2/libssh2	https://github.com/libssh2/libssh2/blob/master/COPYING
libstdcxx	https://gcc.gnu.org/	https://gcc.gnu.org/onlinedocs/libstdc++/manual/license.html

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
libtiff	https://www.libtiff.org/	https://www.libtiff.org/misc.html
libtrust	https://github.com/docker/libtrust	https://github.com/docker/libtrust/blob/master/LICENSE
libuuid	https://sourceforge.net/projects/libuuid/	https://sourceforge.net/p/libuuid/code/ci/master/tree/COPYING
libuv	https://github.com/libuv/libuv	https://github.com/libuv/libuv/blob/v1.x/LICENSE
libwebp	https://chromium.googlesource.com/?format=HTML	https://chromium.googlesource.com/chromium/src/third_party/+/refs/heads/main/libwebp/LICENSE
libwebp-base	https://chromium.googlesource.com/?format=HTML	https://chromium.googlesource.com/chromium/src/third_party/+/refs/heads/main/libwebp/LICENSE
libxc	https://www.tddft.org/programs/libxc/	https://gitlab.com/libxc/libxc/-/blob/master/COPYING
libxcb	https://xcb.freedesktop.org	https://github.com/freedesktop/xcb-libxcb/blob/master/COPYING
libxcrypt	https://github.com/besser82/libxcrypt	https://github.com/besser82/libxcrypt?tab=LGPL-2.1-1-ov-file#readme
libxml2	https://git.gnome.org/browse/libxml2/	https://gitlab.gnome.org/GNOME/libxml2/-/blob/master/Copyright
libxmlsec1	https://github.com/lsh123/xmlsec	https://github.com/lsh123/xmlsec/blob/master/README
libxslt	https://gitlab.gnome.org/GNOME/libxslt/-/wikis/home	https://gitlab.gnome.org/GNOME/libxslt/-/blob/master/Copyright

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
libzlib	https://zlib.net	https://zlib.net/zlib_license.html
lightning-utilities	https://github.com/Lightning-AI/utilities	https://github.com/Lightning-AI/utilities?tab=Apache-2.0-1-ov-file#readme
lime	https://github.com/marcotcr/lime	https://github.com/marcotcr/lime/blob/master/LICENSE
linear-operator	https://github.com/cornellius-gp/linear_operator	https://github.com/cornellius-gp/linear_operator?tab=MIT-1-ov-file#readme
lit	http://llvm.org	https://releases.llvm.org/14.0.0/LICENSE.TXT
llvm-openmp	https://github.com/llvm-mirror/openmp	https://github.com/llvm-mirror/openmp/blob/master/LICENSE.txt
llvmlite	http://llvmlite.readthedocs.io	https://github.com/numba/llvmlite/blob/main/LICENSE
loader-utils	https://github.com/webpack/loader-utils	https://github.com/webpack/loader-utils/raw/master/LICENSE
logistro	https://github.com/geopozo/logistro	https://github.com/geopozo/logistro?tab=MIT-1-ov-file#readme
logomaker	https://logomaker.readthedocs.io/en/latest/	https://github.com/jbkinney/logomaker/blob/master/LICENSE
logrus	https://github.com/sirupsen/logrus	https://github.com/sirupsen/logrus/blob/master/LICENSE
logrus-airbrake-hook.v2	https://gopkg.in/gemnasium/logrus-airbrake-hook.v2	https://github.com/gemnasium/logrus-airbrake-hook/blob/v2.1.2/LICENSE
lxml	https://lxml.de	https://lxml.de/index.html#license

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
lz4-c	https://www.lz4.org/	https://github.com/lz4/lz4/blob/dev/LICENSE
lzo	https://www.oberhumer.com/opensource/lzo/	https://www.oberhumer.com/opensource/gpl.html
mamba	https://github.com/mamba-org/mamba	https://github.com/mamba-org/mamba?tab=BSD-3-Clause-1-ov-f readme
markdown	https://github.com/evilstreak/markdown-js	https://www.opensource.org/licenses/mit-license.php
markdown-it-py	https://github.com/executablebooks/markdown-it-py	https://github.com/executablebooks/markdown-it-py/blob/master/LICENSE
markupsafe	https://github.com/pallets/markupsafe/tree/main/src/markupsafe	https://github.com/pallets/markupsafe/tree/main?tab=BSD-3-Clause-1-ov-f
MarkupSafe	https://palletsprojects.com/p/markupsafe/	https://github.com/pallets/markupsafe/blob/main/LICENSE.rst
mashumaro	https://github.com/Fatal1ty/mashumaro	https://github.com/Fatal1ty/mashumaro
matplotlib	https://matplotlib.org	https://matplotlib.org/stable/users/project/license.html
matplotlib-base	https://matplotlib.org	https://matplotlib.org/stable/users/project/license.html
matplotlib-inline	https://github.com/ipython/matplotlib-inline	https://github.com/ipython/matplotlib-inline/blob/master/LICENSE
mda-xdrlib	https://github.com/MDAnalysis/mda-xdrlib	https://github.com/MDAnalysis/mda-xdrlib/blob/main/LICENSE

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
mda_xdrlib	https://github.com/MDAnalysis/mda-xdrlib	https://github.com/MDAnalysis/mda-xdrlib?tab=License-1-ov-file#readme
mdtraj	https://www.mdtraj.org/	https://www.mdtraj.org/1.9.8.dev0/index.html#license
mdurl	https://github.com/markdown-it/mdurl	https://github.com/markdown-it/mdurl/blob/master/LICENSE
menuinst	https://github.com/conda/menuinst	https://github.com/conda/menuinst/blob/main/LICENSE.txt
mergo	https://github.com/imdario/mergo	https://github.com/imdario/mergo/blob/master/LICENSE
mistune	https://github.com/lepture/mistune	https://github.com/lepture/mistune/blob/master/LICENSE
mkl	https://github.com/rust-math/intel-mkl-src	https://github.com/rust-math/intel-mkl-src/blob/master/License.txt
mkl-fft	https://github.com/IntelPython/mkl_fft	https://github.com/IntelPython/mkl_fft/blob/master/LICENSE.txt
mkl-random	https://github.com/IntelPython/mkl_random	https://github.com/IntelPython/mkl_random/blob/master/LICENSE.txt
mkl-service	https://github.com/IntelPython/mkl-service	https://github.com/IntelPython/mkl-service/blob/master/LICENSE.txt
mkl_fft	http://github.com/IntelPython/mkl_fft	https://github.com/IntelPython/mkl_fft?tab=BSD-3-Clause-1-ov-file#readme

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
mkl_random	http://github.com/IntelPython/mkl_random	https://github.com/IntelPython/mkl_random?tab=BSD-3-Clause-1-ov-f readme
ml-dtypes	https://github.com/jax-ml/ml_dtypes	https://github.com/jax-ml/ml_dtypes/blob/main/LICENSE
ml_dtypes	https://github.com/jax-ml/ml_dtypes	https://github.com/jax-ml/ml_dtypes/blob/main/LICENSE
modelcif	https://github.com/ihmwg/python-modelcif	https://github.com/ihmwg/python-modelcif/blob/main/LICENSE
molecupy	https://molecupy.readthedocs.io/en/latest/	https://github.com/samirelanduk/atomium/blob/master/LICENSE
moment	https://github.com/moment/moment	https://github.com/moment/moment/raw/master/LICENSE
moment-precise-range-plugin	https://github.com/codebox/moment-precise-range	https://github.com/codebox/moment-precise-range raw/master/LICENSE.md
more-itertools	https://github.com/more-itertools/more-itertools	https://github.com/more-itertools/more-itertools/blob/master/LICENSE
mpiplus	https://github.com/choderalab/mpiplus	https://github.com/choderalab/mpiplus/blob/main/LICENSE
mpmath	http://mpmath.org/	https://github.com/mpmath/mpmath/blob/master/LICENSE
mrcfile	https://github.com/ccpem/mrcfile	https://github.com/ccpem/mrcfile/blob/master/LICENSE.txt

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
msgpack	https://msgpack.org/	https://github.com/msgpack/msgpack-python/blob/main/COPYING
multidict	https://github.com/aio-libs/multidict	https://github.com/aio-libs/multidict/blob/master/LICENSE
multierr	https://go.uber.org/multierr	https://github.com/uber-go/multierr/blob/master/LICENSE.txt
multiprocess	https://github.com/uqfoundation/multiprocess	https://github.com/uqfoundation/multiprocess/blob/master/LICENSE
munkres	https://software.clapper.org/munkres/	https://github.com/bmc/munkres/blob/master/LICENSE.md
myesui uuid	https://github.com/myesui/uuid	https://pkg.go.dev/github.com/myesui/uuid?tab=licenses
namex	https://github.com/bcgov/namex	https://github.com/bcgov/namex/blob/main/LICENSE
narwhals	https://github.com/narwhals-dev/narwhals	https://github.com/narwhals-dev/narwhals?tab=MIT-1-ov-file#readme
nbclassic	http://jupyter.org	https://github.com/jupyter/nbclassic/blob/main/LICENSE
nbclient	https://jupyter.org	https://github.com/jupyter/nbclient/blob/main/LICENSE
nbconvert	https://jupyter.org	https://github.com/jupyter/nbconvert/blob/main/LICENSE
nbformat	http://jupyter.org	https://github.com/jupyter/nbformat/blob/main/LICENSE

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
ncurses	https://invisible-island.net/ncurses/	https://invisible-island.net/ncurses/ncurses-license.html
ndfes	https://rutgerslbsr.gitlab.io/fe-toolkit/	https://gitlab.com/RutgersLBSR/fe-toolkit/-/blob/master/LICENSE
ndindex	https://quansight-labs.github.io/ndindex/	https://github.com/Quansight-Labs/ndindex?tab=MIT-1-ov-file#readme
nest-asyncio	https://github.com/erdewit/nest_asyncio	https://github.com/erdewit/nest_asyncio/blob/master/LICENSE
nest_asyncio	https://github.com/erdewit/nest_asyncio	https://github.com/erdewit/nest_asyncio?tab=BSD-2-Clause-1-ov-f readme
netcdf-fortran	https://www.unidata.ucar.edu/software/netcdf/	https://www.unidata.ucar.edu/software/netcdf/copyright.html
netCDF4	http://github.com/Unidata/netcdf4-python	https://github.com/Unidata/netcdf4-python/blob/master/LICENSE
nettle	https://git.lysator.liu.se/nettle/nettle	https://git.lysator.liu.se/nettle/nettle/-/blob/master/AUTHORS?ref_type=heads
networkx	https://networkx.org	https://github.com/networkx/networkx/blob/main/LICENSE.txt
nfpm	https://github.com/goreleaser/nfpm	https://github.com/goreleaser/nfpm/blob/master/LICENSE.md
ng-tags-input	https://github.com/mbenford/ngTagsInput	https://github.com/mbenford/ngTagsInput
ng-toast	https://github.com/tameraydin/ngToast	https://github.com/tameraydin/ngToast

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
ngdraggable	https://github.com/fatlinesofcode/ngDraggable	https://github.com/fatlinesofcode/ngDraggable/raw/master/LICENSE.txt
ngVue	https://github.com/ngVue/ngVue	https://github.com/ngVue/ngVue/raw/master/LICENSE
nh3	https://nh3.readthedocs.io/en/latest/	https://github.com/messense/nh3/blob/main/LICENSE
nlohmann_json	https://github.com/nlohmann/json	https://github.com/nlohmann/json?tab=MIT-1-ov-file#readme
nlopt	https://nlopt.readthedocs.io/en/latest/NLopt_Python_Reference/	https://nlopt.readthedocs.io/en/latest/NLopt_License_and_Copyright/
nodejs	https://nodejs.org/en/	https://raw.githubusercontent.com/nodejs/node/master/LICENSE
nomkl	https://github.com/conda-forge/nomkl-feedstock	https://github.com/conda-forge/nomkl-feedstock/blob/main/LICENSE.txt
notebook	http://jupyter.org	https://github.com/jupyter/notebook/blob/main/LICENSE
notebook-shim	https://github.com/jupyter/notebook_shim	https://github.com/jupyter/notebook_shim
notebook_shim	http://jupyter.org	https://github.com/jupyter/notebook_shim/blob/main/LICENSE
numba	https://numba.pydata.org	https://github.com/numba/numba/blob/main/LICENSE
numcpus	https://github.com/tklauser/numcpus	https://github.com/tklauser/numcpus/blob/master/LICENSE

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
numexpr	https://github.com/pydata/numexpr	https://github.com/pydata/numexpr/blob/master/LICENSE.txt
numpy	https://numpy.org	https://github.com/numpy/numpy/blob/main/LICENSE.txt
numpy-base	https://numpy.org	https://github.com/numpy/numpy/blob/main/LICENSE.txt
numpydoc	https://numpydoc.readthedocs.io/en/latest/index.html	https://github.com/numpy/numpydoc/blob/main/LICENSE.txt
nvidia-cublas-cu11	https://developer.nvidia.com/cuda-zone	https://docs.nvidia.com/cuda/eula/index.html
nvidia-cublas-cu12	https://developer.nvidia.com/cuda-zone	https://docs.nvidia.com/cuda/eula/index.html
nvidia-cuda-cupti-cu11	https://developer.nvidia.com/cuda-zone	https://docs.nvidia.com/cuda/eula/index.html
nvidia-cuda-cupti-cu12	https://developer.nvidia.com/cuda-zone	https://docs.nvidia.com/cuda/eula/index.html
nvidia-cuda-nvrtc-cu11	https://developer.nvidia.com/cuda-zone	https://docs.nvidia.com/cuda/eula/index.html
nvidia-cuda-nvrtc-cu12	https://developer.nvidia.com/cuda-zone	https://docs.nvidia.com/cuda/eula/index.html
nvidia-cuda-runtime-cu11	https://developer.nvidia.com/cuda-zone	https://docs.nvidia.com/cuda/eula/index.html
nvidia-cuda-runtime-cu12	https://developer.nvidia.com/cuda-zone	https://docs.nvidia.com/cuda/eula/index.html
nvidia-cudnn-cu11	https://developer.nvidia.com/cuda-zone	https://docs.nvidia.com/deeplearning/cudnn/sla/index.html
nvidia-cudnn-cu12	https://developer.nvidia.com/cuda-zone	https://docs.nvidia.com/deeplearning/cudnn/sla/index.html

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
nvidia-cuEquivariance	https://github.com/NVIDIA/cuEquivariance	https://github.com/NVIDIA/cuEquivariance/blob/main/LICENSES/LICENSE
nvidia-cufft-cu11	https://developer.nvidia.com/cuda-zone	https://docs.nvidia.com/cuda/eula/index.html
nvidia-cufft-cu12	https://developer.nvidia.com/cuda-zone	https://docs.nvidia.com/cuda/eula/index.html
nvidia-cufile-cu12	https://developer.nvidia.com/cuda-zone	https://docs.nvidia.com/cuda/eula/index.html#
nvidia-curand-cu11	https://developer.nvidia.com/cuda-zone	https://docs.nvidia.com/cuda/eula/index.html
nvidia-curand-cu12	https://developer.nvidia.com/cuda-zone	https://docs.nvidia.com/cuda/eula/index.html
nvidia-cusolver-cu11	https://developer.nvidia.com/cuda-zone	https://docs.nvidia.com/cuda/eula/index.html
nvidia-cusolver-cu12	https://developer.nvidia.com/cuda-zone	https://docs.nvidia.com/cuda/eula/index.html
nvidia-cusparse-cu11	https://developer.nvidia.com/cuda-zone	https://docs.nvidia.com/cuda/eula/index.html
nvidia-cusparse-cu12	https://developer.nvidia.com/cuda-zone	https://docs.nvidia.com/cuda/eula/index.html
nvidia-cusparselt-cu12	https://docs.nvidia.com/cuda/cusparselt/index.html	https://docs.nvidia.com/cuda/cusparselt/license.html
nvidia-ml-py	https://pypi.org/project/nvidia-ml-py/	https://pypi.org/project/nvidia-ml-py/
nvidia-nccl-cu11	https://developer.nvidia.com/cuda-zone	https://docs.nvidia.com/hpc-sdk/eula/index.html
nvidia-nccl-cu12	https://developer.nvidia.com/cuda-zone	https://docs.nvidia.com/hpc-sdk/eula/index.html
nvidia-nvjitlink-cu12	https://developer.nvidia.com/cuda-zone	https://docs.nvidia.com/cuda/eula/index.html

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
nvidia-nvshmem-cu12	https://developer.nvidia.com/cuda-zone	https://docs.nvidia.com/cuda/eula/index.html
nvidia-nvtx-cu11	https://developer.nvidia.com/cuda-zone	https://docs.nvidia.com/cuda/eula/index.html
nvidia-nvtx-cu12	https://developer.nvidia.com/cuda-zone	https://docs.nvidia.com/cuda/eula/index.html
Oat++	https://oatpp.io/	https://github.com/oatpp/oatpp/blob/master/LICENSE
oauthlib	https://github.com/oauthlib/oauthlib	https://github.com/oauthlib/oauthlib/blob/master/LICENSE
ocl-icd	https://github.com/OCL-dev/ocl-icd	https://github.com/OCL-dev/ocl-icd/blob/master/COPYING
ocl-icd-system	https://github.com/conda-forge/ocl-icd-system-feedst	https://github.com/conda-forge/ocl-icd-system-feedst blob/main/LICENSE.txt
olefile	https://www.decalage.info/python/olefileio	https://olefile.readthedocs.io/en/latest/License.html
omegaconf	https://github.com/omry/omegaconf	https://github.com/omry/omegaconf/blob/master/LICENSE
OmegaFold	https://github.com/HeliXonProtein/OmegaFold/tree/main	https://github.com/HeliXonProtein/OmegaFold/blob/main/LICENSE
omnicanvas	https://omnicanvas.readthedocs.io/en/latest/	https://github.com/samirelanduk/omnicanvas/blob/master/LICENSE
OpenFF	https://openforcefield.org/	https://github.com/openforcefield/openff-forcefields
openff-amber-ff-ports	https://github.com/openforcefield/openff-amber-ff-ports	https://github.com/openforcefield/openff-amber-ff-ports blob/main/LICENSE

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
openff-forcefields	https://openforcefield.org	https://github.com/openforcefield/openff-forcefields/blob/master/LICENSE
openff-fragmenter	https://github.com/openforcefield/openff-fragmenter	https://github.com/openforcefield/openff-fragmenter/blob/main/LICENSE
openff-interchange	https://github.com/openforcefield/openff-interchange	https://github.com/openforcefield/openff-interchange/blob/main/LICENSE
openff-models	https://github.com/openforcefield/openff-models	https://github.com/openforcefield/openff-models/blob/main/LICENSE
openff-nagl	https://docs.openforcefield.org/projects/nagl/en/latest/?badge=latest	https://github.com/openforcefield/openff-nagl/blob/main/LICENSE
openff-nagl-models	https://github.com/openforcefield/openff-nagl-models	https://github.com/openforcefield/openff-nagl-models/blob/main/LICENSE
openff-toolkit	https://openforcefield.org	https://github.com/openforcefield/openff-toolkit/blob/master/LICENSE
openff-toolkit-base	https://openforcefield.org	https://github.com/openforcefield/openff-toolkit/blob/master/LICENSE
openff-units	https://github.com/openforcefield/openff-units	https://github.com/openforcefield/openff-units/blob/main/LICENSE
openff-utilities	https://github.com/openforcefield/openff-utilities	https://github.com/openforcefield/openff-utilities/blob/main/LICENSE
openjpeg	https://github.com/uclouvain/openjpeg	https://github.com/uclouvain/openjpeg/blob/master/LICENSE

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
openldap	https://www.openldap.org/software/repo.html	https://git.openldap.org/openldap/openldap/-/blob/master/LICENSE
OpenMM	https://openmm.org	https://docs.openmm.org/7.1.0/userguide/library.html#license
openmmtools	https://github.com/choderalab/openmmtools	https://github.com/choderalab/openmmtools/blob/main/LICENSE
openmoltools	https://github.com/choderalab/openmoltools	https://github.com/choderalab/openmoltools/blob/master/LICENSE
openpyxl	https://openpyxl.readthedocs.io/en/stable/	https://foss.heptapod.net/openpyxl/openpyxl/-/blob/branch/3.0/LICENCE.rst
openssl	https://www.openssl.org	https://github.com/openssl/openssl/blob/master/LICENSE.txt
opt-einsum	https://github.com/dgasmith/opt_einsum	https://github.com/dgasmith/opt_einsum/blob/master/LICENSE
opt_einsum	https://github.com/dgasmith/opt_einsum	https://github.com/dgasmith/opt_einsum/blob/main/LICENSE
OptKing	https://github.com/psi-rking/optking	https://github.com/psi-rking/optking/blob/master/LICENSE
optree	https://optree.readthedocs.io/en/latest/	https://github.com/metaopt/optree/blob/main/LICENSE
oscrypto	https://github.com/wbond/oscrypto	https://github.com/wbond/oscrypto/blob/master/LICENSE
overrides	https://github.com/mkorpela/overrides	https://github.com/mkorpela/overrides/blob/main/LICENSE

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
packaging	https://github.com/pypa/packaging	https://github.com/pypa/packaging/blob/main/LICENSE
packmol	https://leandro.iqm.unicamp.br/m3g/packmol/home.shtml	https://github.com/m3g/packmol/blob/master/LICENSE
packmol_memgen	http://cpclab.uni-duesseldorf.de/	https://gitlab.com/cbjh/memgen/py-memgen/-/blob/main/LICENSE
pandas	https://pandas.pydata.org	https://github.com/pandas-dev/pandas/blob/main/LICENSE
pandocfilters	http://github.com/jgm/pandocfilters	https://github.com/jgm/pandocfilters/blob/master/LICENSE
panedr	https://github.com/MDAnalysis/panedr	https://github.com/MDAnalysis/panedr/blob/master/LICENSE.txt
pango	https://pango.gnome.org	https://gitlab.gnome.org/GNOME/pango/-/blob/main/COPYING
ParmEd	https://parmed.github.io/ParmEd/html/index.html	https://github.com/ParmEd/ParmEd#license
parser	https://github.com/typescript-eslint/typescript-eslint	https://github.com/typescript-eslint/typescript-eslint/raw/master/LICENSE
parso	https://parso.readthedocs.io/en/latest/	https://github.com/davidhalter/parso/blob/master/LICENSE.txt
pathos	https://github.com/uqfoundation/pathos	https://github.com/uqfoundation/pathos/blob/master/LICENSE
patsy	https://patsy.readthedocs.io/en/latest/	https://github.com/pydata/patsy/blob/master/LICENSE.txt

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
pbkdf2	https://golang.org/x/crypto/pbkdf2	https://cs.opensource.google/go/x/crypto/+/master:LICENSE
pbr	https://docs.openstack.org/pbr/latest/	https://opendev.org/openstack/pbr/src/branch/master/LICENSE
pcmsolver	https://pcmsolver.readthedocs.io/en/stable/	https://github.com/PCMSolver/pcmsolver/blob/master/LICENSE
pcre	https://www.pcre.org	https://www.pcre.org/licence.txt
pcre2	https://www.pcre.org	https://github.com/PCRE2Project/pcre2/blob/master/LICENCE
pdb4amber	https://github.com/Amber-MD/pdb4amber	https://github.com/Amber-MD/pdb4amber/blob/master/LICENSE
pdbfixer	https://github.com/openmm/pdbfixer	https://github.com/openmm/pdbfixer/blob/master/LICENSE
pexpect	https://pexpect.readthedocs.io/	https://github.com/pexpect/pexpect/blob/master/LICENSE
pgconn	https://github.com/jackc/pgconn	https://github.com/jackc/pgconn/blob/master/LICENSE
pgio	https://github.com/jackc/pgio	https://github.com/jackc/pgio/blob/master/LICENSE
pgpassfile	https://github.com/jackc/pgpassfile	https://github.com/jackc/pgpassfile/blob/master/LICENSE
pgproto3	https://github.com/jackc/pgproto3/v2	https://github.com/jackc/pgproto3/blob/master/LICENSE
pgtype	https://github.com/jackc/pgtype	https://github.com/jackc/pgtype/blob/master/LICENSE
pgx	https://github.com/jackc/pgx/v4	https://github.com/jackc/pgx/blob/v4.0.1/LICENSE

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
phonopy	https://github.com/phonopy/phono3py	https://github.com/phonopy/phono3py/blob/develop/LICENSE
pickleshare	https://github.com/pickleshare/pickleshare	https://github.com/pickleshare/pickleshare/blob/master/LICENSE
Pillow	https://python-pillow.org	https://github.com/python-pillow/Pillow/blob/main/LICENSE
Pint	https://pint.readthedocs.io/en/stable/	https://github.com/hgrecco/pint/blob/master/LICENSE
pip	https://pip.pypa.io/	https://github.com/pypa/pip/blob/main/LICENSE.txt
pip-licenses	https://github.com/raimon49/pip-licenses	https://github.com/raimon49/pip-licenses/blob/master/LICENSE
pixman	https://pixman.org	https://cgit.freedesktop.org/pixman/tree/COPYING
pkginfo	https://launchpad.net/pkginfo	https://bazaar.launchpad.net/~tseaver/pkginfo/trunk/view/head:/LICENSE.txt
pkgutil-resolve-name	https://github.com/graingert/pkgutil-resolve-name	https://github.com/graingert/pkgutil-resolve-name?tab=License-1-ov-file#readme
pkgutil_resolve_name	https://github.com/graingert/pkgutil-resolve-name	https://github.com/graingert/pkgutil-resolve-name?tab=License-1-ov-file#readme
platformdirs	https://github.com/platformdirs/platformdirs	https://github.com/platformdirs/platformdirs/blob/main/LICENSE

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
plotly	https://plotly.com/python/	https://github.com/plotly/plotly.py/blob/master/LICENSE.txt
plotly-orca	https://github.com/plotly/orca	https://github.com/plotly/orca/blob/master/LICENSE
plotly.js	https://github.com/plotly/plotly.js	https://github.com/plotly/plotly.js/raw/master/LICENSE
pluggy	https://pluggy.readthedocs.io/en/stable/index.html	https://github.com/pytest-dev/pluggy/blob/main/LICENSE
pooch	https://github.com/fatiando/pooch	https://github.com/fatiando/pooch/blob/main/LICENSE.txt
portalocker	https://github.com/WoLpH/portalocker	https://github.com/wolph/portalocker/blob/develop/LICENSE
pox	https://github.com/uqfoundation/pox	https://github.com/uqfoundation/pox/blob/master/LICENSE
ppft	https://github.com/uqfoundation/ppft	https://github.com/uqfoundation/ppft/blob/master/LICENSE
pq	https://github.com/lib/pq	https://github.com/lib/pq/blob/master/LICENSE.md
ProDy	http://www.csb.pitt.edu/ProDy	https://github.com/prody/ProDy/blob/master/LICENSE.rst
prometheus-client	https://github.com/prometheus/client_python	https://github.com/prometheus/client_python/blob/master/LICENSE
prometheus_client	https://github.com/prometheus/client_python	https://github.com/prometheus/client_python/blob/master/LICENSE
prompt-toolkit	https://python-prompt-toolkit.readthedocs.io/en/stable/	https://github.com/prompt-toolkit/python-prompt-toolkit/blob/master/LICENSE

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
prompt_toolkit	https://github.com/prompt-toolkit/python-prompt-toolki	https://github.com/prompt-toolkit/python-prompt-toolki tab=BSD-3-Clause-1-ov-f readme
propcache	https://github.com/aio-libs/propcache	https://github.com/aio-libs/propcache?tab=Apache-2.0-1-ov-file#readme
protobuf	https://google.golang.org/protobuf	https://github.com/protocolbuffers/protobuf-go/blob/master/LICENSE
psi4	https://psicode.org	https://github.com/psi4/psi4/blob/master/COPYING.LESSER
psutil	https://psutil.readthedocs.io/en/latest/	https://github.com/giampaolo/psutil/blob/master/LICENSE
psycopg2	https://psycopg.org/	https://github.com/psycopg/psycopg2/blob/master/LICENSE
PTable	https://github.com/kxxoling/PTable	https://github.com/kxxoling/PTable/blob/master/COPYING
pthread-stubs	https://xcb.freedesktop.org	https://github.com/freedesktop/xcb-pthread-stubs/blob/master/COPYING
ptyprocess	https://ptyprocess.readthedocs.io/en/latest/	https://github.com/pexpect/ptyprocess/blob/master/LICENSE
pure-eval	http://github.com/alexmojaki/pure_eval	http://github.com/alexmojaki/pure_eval/blob/master/LICENSE.txt
pure_eval	http://github.com/alexmojaki/pure_eval	https://github.com/alexmojaki/pure_eval?tab=MIT-1-ov-file#readme

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
py	https://py.readthedocs.io/en/latest/	https://github.com/pytest-dev/py/blob/master/LICENSE
py-cpuinfo	https://github.com/workhorsy/py-cpuinfo	https://github.com/workhorsy/py-cpuinfo/blob/master/LICENSE
pyasn1	https://github.com/etingof/pyasn1	https://github.com/etingof/pyasn1/blob/master/LICENSE.rst
pyasn1-modules	https://github.com/etingof/pyasn1-modules	https://github.com/etingof/pyasn1-modules/blob/master/LICENSE.txt
pyasn1_modules	https://pyasn1.readthedocs.io/en/latest/contents.html	https://github.com/pyasn1/pyasn1-modules?tab=BSD-2-Clause-1-ov-file#readme
pybind11	http://pybind11.readthedocs.io/en/stable/	https://github.com/pybind/pybind11/blob/master/LICENSE
pybind11-abi	https://github.com/pybind/pybind11	https://github.com/pybind/pybind11/blob/master/LICENSE
pybind11_global	http://pybind11.readthedocs.io/en/stable/	https://github.com/pybind/pybind11/blob/master/LICENSE
pycairo	https://pycairo.readthedocs.io/en/latest/index.html	https://github.com/pygobject/pycairo/blob/master/COPYING
pycosat	https://github.com/conda/pycosat	https://github.com/conda/pycosat/blob/main/LICENSE
pycparser	https://github.com/eliben/pycparser	https://github.com/eliben/pycparser/blob/master/LICENSE
pydantic	https://pydantic-docs.helpmanual.io	https://github.com/samuelcolvin/pydantic/blob/master/LICENSE

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
pydantic-core	https://github.com/pydantic/pydantic-core	https://github.com/pydantic/pydantic-core
pyedr	https://github.com/MDAnalysis/panedr	https://github.com/MDAnalysis/panedr/blob/master/LICENSE.txt
pyEMMA	http://github.com/markovmodel/PyEMMA	https://github.com/markovmodel/PyEMMA/blob/devel/LICENSE.txt
Pygments	https://pygments.org	https://github.com/pygments/pygments/blob/master/LICENSE
pygments	https://pygments.org/	https://github.com/pygments/pygments?tab=BSD-2-Clause-1-ov-file#readme
pygraphviz	https://pygraphviz.github.io	https://github.com/pygraphviz/pygraphviz/blob/main/LICENSE
pygtop	https://pygtop.readthedocs.io/en/latest/	https://github.com/samirelanduk/pygtop/blob/master/LICENSE
pyHanko	https://github.com/MatthiasValvekens/pyHanko	https://github.com/MatthiasValvekens/pyHanko/blob/master/LICENSE
pyhanko-certvalidator	https://github.com/MatthiasValvekens/certvalidator	https://github.com/MatthiasValvekens/certvalidator/blob/master/LICENSE
PyJWT	https://github.com/jpadilla/pyjwt	https://github.com/jpadilla/pyjwt/blob/master/LICENSE
pymbar	https://pymbar.org	https://github.com/choderalab/pymbar/blob/master/LICENSE
pynvml	https://github.com/gpuopenanalytics/pynvml	https://github.com/gpuopenanalytics/pynvml/blob/master/LICENSE.txt

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
pyOpenSSL	https://pyopenssl.org/	https://github.com/pyca/pyopenssl/blob/main/LICENSE
pyparsing	https://pyparsing-docs.readthedocs.io/en/latest/	https://github.com/pyparsing/pyparsing/blob/master/LICENSE
PyPDF3	https://github.com/sfneal/PyPDF3	https://github.com/sfneal/PyPDF3/blob/master/LICENSE
pyrsistent	http://github.com/tobgu/pyrsistent/	https://github.com/tobgu/pyrsistent/blob/master/LICENSE.mit
pysam	https://github.com/pysam-developers/pysam	https://github.com/pysam-developers/pysam/blob/master/COPYING
PySocks	https://github.com/Anorov/PySocks	https://github.com/Anorov/PySocks/blob/master/LICENSE
pytables	https://www.pytables.org	https://github.com/PyTables/PyTables/blob/master/LICENSE.txt
python	https://www.python.org/	https://docs.python.org/3/license.html
python-bidi	https://github.com/MeirKriheli/python-bidi	https://github.com/MeirKriheli/python-bidi/blob/master/COPYING
python-constraint	https://github.com/python-constraint/python-constraint	https://github.com/python-constraint/python-constraint/blob/master/LICENSE
python-dateutil	https://dateutil.readthedocs.io	https://github.com/dateutil/dateutil/blob/master/LICENSE
python-json-logger	http://github.com/madzak/python-json-logger	https://github.com/madzak/python-json-logger/blob/master/LICENSE

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
python-pptx	https://github.com/scanny/python-pptx	https://github.com/scanny/python-pptx/blob/master/LICENSE
python-ldap	https://www.python-ldap.org/	https://github.com/python-ldap/python-ldap/blob/python-ldap-3.4.0/LICENCE
python3-saml	https://github.com/onelogin/python3-saml	https://github.com/onelogin/python3-saml/blob/master/LICENSE
python_abi	https://github.com/conda-forge/python_abi-feedstock	https://github.com/conda-forge/python_abi-feedstock/blob/main/LICENSE.txt
pytorch-lightning	https://github.com/Lightning-AI/pytorch-lightning	https://github.com/Lightning-AI/pytorch-lightning/blob/master/LICENSE
pytz	https://pythonhosted.org/pytz	https://git.launchpad.net/pytz/tree/LICENSE.txt
pytz-deprecation-shim	https://github.com/pganssle/pytz-deprecation-shim	https://github.com/pganssle/pytz-deprecation-shim/blob/master/LICENSE
PyWavelets	https://github.com/PyWavelets/pywt	https://github.com/PyWavelets/pywt/blob/master/LICENSE
PyYAML	https://pyyaml.org/	https://github.com/yaml/pyyaml/blob/master/LICENSE
pyyml	No longer available	No longer available
pyzmq	https://pyzmq.readthedocs.io/en/latest/	https://github.com/zeromq/pyzmq/blob/main/COPYING.BSD
qcelemental	https://github.com/MolSSI/QCElemental	https://github.com/MolSSI/QCElemental/blob/master/LICENSE

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
qcengine	https://github.com/MolSSI/QCEngine	https://github.com/MolSSI/QCEngine/blob/master/LICENSE
qrcode	https://github.com/lincolnloop/python-qrcode	https://github.com/lincolnloop/python-qrcode/blob/main/LICENSE
ramda	https://github.com/ramda/ramda	https://github.com/ramda/ramda/raw/master/LICENSE.txt
rapidjson	https://rapidjson.org/	https://github.com/Tencent/rapidjson/blob/master/license.txt
rdkit	https://www.rdkit.org	https://github.com/rdkit/rdkit/blob/master/license.txt
re2	https://github.com/google/re2	https://github.com/google/re2/blob/main/LICENSE
readme-renderer	https://github.com/pypa/readme_renderer	https://github.com/pypa/readme_renderer/blob/main/LICENSE
redis-py	https://github.com/andymccurdy/redis-py	https://github.com/redis/redis-py/blob/master/LICENSE
referencing	https://github.com/python-jsonschema/referencing	https://github.com/python-jsonschema/referencing
regex	https://github.com/mrabarnett/mrab-regex	https://github.com/mrabarnett/mrab-regex/blob/hg/LICENSE.txt
reportlab	https://www.reportlab.com	https://www.reportlab.com/dev/docs/faq/#1.3
reproc	https://github.com/DaanDeMeyer/reproc	https://github.com/DaanDeMeyer/reproc/blob/main/LICENSE
reproc-cpp	https://github.com/DaanDeMeyer/reproc	https://github.com/DaanDeMeyer/reproc/blob/main/LICENSE

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
requests	https://requests.readthedocs.io	https://github.com/psf/requests/blob/main/LICENSE
requests-oauthlib	https://github.com/requests/requests-oauthlib	https://github.com/requests/requests-oauthlib/blob/master/LICENSE
requests-toolbelt	https://toolbelt.readthedocs.org	https://github.com/requests/toolbelt/blob/master/LICENSE
resumable	https://github.com/stevvooe/resumable	https://github.com/stevvooe/resumable/blob/master/LICENSE
retrying	https://github.com/rholder/retrying	https://github.com/rholder/retrying/blob/master/LICENSE
rfc3339-validator	https://github.com/naimetti/rfc3339-validator	https://github.com/naimetti/rfc3339-validator/blob/master/LICENSE
rfc3339_validator	https://github.com/naimetti/rfc3339-validator	https://github.com/naimetti/rfc3339-validator?tab=MIT-1-ov-file#readme
rfc3986	https://rfc3986.readthedocs.io/en/latest/	https://github.com/python-hyper/rfc3986/blob/main/LICENSE
rfc3986-validator	https://github.com/naimetti/rfc3986-validator	https://github.com/naimetti/rfc3986-validator/blob/master/LICENSE
rfc3987-syntax	https://github.com/willynilly/rfc3987-syntax	https://github.com/willynilly/rfc3987-syntax?tab=MIT-1-ov-file#readme
rich	https://github.com/Textualize/rich	https://github.com/Textualize/rich/blob/master/LICENSE

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
rich-argparse	https://github.com/hamdanal/rich-argparse	https://github.com/hamdanal/rich-argparse/blob/main/LICENSE
rlPyCairo	https://pypi.org/project/rlPyCairo/	https://hg.reportlab.com/hg-public/rlPyCairo/file/tip/rlPyCairo/LICENSE.txt
RocksDict	https://github.com/rocksdict/RocksDict	https://github.com/rocksdict/RocksDict?tab=MIT-1-ov-file#readme
roman-numerals	https://github.com/AA-Turner/roman-numerals	https://github.com/AA-Turner/roman-numerals?tab=License-1-ov-file#readme
roman-numerals-py	https://github.com/AA-Turner/roman-numerals	https://github.com/AA-Turner/roman-numerals?tab=License-1-ov-file#readme
rotary-embedding-torch	https://github.com/lucidrains/rotary-embedding-tor	https://github.com/lucidrains/rotary-embedding-tor blob/main/LICENSE
rpds-py	https://github.com/crate-py/rpds	https://github.com/crate-py/rpds
rpmpack	https://github.com/google/rpmpack	https://github.com/google/rpmpack/blob/master/LICENSE
rsa	https://stuvel.eu/rsa	https://github.com/sybrenstuvel/python-rsa/blob/main/LICENSE
ruamel-yaml	https://sourceforge.net/p/ruamel-yaml/code/ci/default/tree/	https://sourceforge.net/p/ruamel-yaml/code/ci/default/tree/LICENSE
ruamel.yaml.clib	https://sourceforge.net/p/ruamel-yaml-clib/code/ci/default/tree/	https://sourceforge.net/p/ruamel-yaml-clib/code/ci/default/tree/LICENSE

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
s3transfer	https://github.com/boto/s3transfer	https://github.com/boto/s3transfer/blob/develop/LICENSE.txt
sacrebleu	https://github.com/mjpost/sacrebleu	https://github.com/mjpost/sacrebleu/blob/master/LICENSE.txt
sapiens	https://github.com/Merck/Sapiens	https://github.com/Merck/Sapiens/blob/main/LICENSE
sasl	https://mellium.im/sasl	https://github.com/mellium/sasl/blob/main/LICENSE
scikit-gstat	https://mmaelicke.github.io/scikit-gstat/	https://github.com/mmaelicke/scikit-gstat/blob/main/LICENSE
scikit-image	https://scikit-image.org	https://scikit-image.org/docs/dev/license.html
scikit-learn	https://scikit-learn.org/stable/	https://github.com/scikit-learn/scikit-learn/blob/main/COPYING
scikit-learn-extra	https://github.com/scikit-learn-contrib/scikit-learn-extra	https://github.com/scikit-learn-contrib/scikit-learn-extra/blob/main/LICENSE
scikit-network	https://github.com/sknetwork-team/scikit-network	https://github.com/sknetwork-team/scikit-network?tab=License-1-ov-file#readme
scipy	https://scipy.org	https://github.com/scipy/scipy/blob/main/LICENSE.txt
seaborn	https://seaborn.pydata.org	https://github.com/mwaskom/seaborn/blob/master/LICENSE
seaborn-base	https://seaborn.pydata.org	https://github.com/mwaskom/seaborn/blob/master/LICENSE

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
semver	https://github.com/Masterminds/semver/v3	https://github.com/Masterminds/semver/blob/v3.0.3/LICENSE.txt
Send2Trash	https://github.com/arsenetar/send2trash	https://github.com/arsenetar/send2trash/blob/master/LICENSE
sentry-sdk	https://github.com/getsentry/sentry-python	https://github.com/getsentry/sentry-python/blob/master/LICENSE
setproctitle	https://github.com/dvarrazzo/py-setproctitle	https://github.com/dvarrazzo/py-setproctitle/blob/master/LICENSE
setuptools	https://github.com/pypa/setuptools	https://github.com/pypa/setuptools/blob/main/LICENSE
setuptools-scm	https://github.com/pypa/setuptools_scm/	https://github.com/pypa/setuptools_scm/blob/main/LICENSE
sh	https://github.com/amoffat/sh	https://github.com/amoffat/sh/blob/develop/LICENSE.txt
shellingham	https://github.com/sarugaku/shellingham	https://github.com/sarugaku/shellingham/blob/master/LICENSE
simdjson	https://github.com/simdjson/simdjson/tree/master	https://github.com/simdjson/simdjson/blob/master/LICENSE
simint	https://www.bennyp.org/research/simint/	https://github.com/simint-chem/simint-generator/blob/master/LICENSE
simplejson	https://github.com/simplejson/simplejson	https://github.com/simplejson/simplejson/blob/master/LICENSE.txt
six	https://github.com/benjaminp/six	https://github.com/benjaminp/six/blob/master/LICENSE

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
smirnoff99frosst	https://github.com/openforcefield/smirnoff99frosst	https://github.com/openforcefield/smirnoff99Frosst/blob/master/LICENSE
smmap	https://github.com/gitpython-developers/smmap	https://github.com/gitpython-developers/smmap/blob/master/LICENSE
snappy	https://github.com/google/snappy	https://github.com/google/snappy/blob/main/COPYING
sniffio	https://github.com/python-trio/sniffio	https://github.com/python-trio/sniffio/blob/master/LICENSE.MIT
snowballstemmer	https://github.com/snowballstem/snowball	https://github.com/snowballstem/snowball/blob/master/COPYING
soupsieve	https://github.com/facelessuser/soupsieve	https://github.com/facelessuser/soupsieve/blob/main/LICENSE.md
spdlog	https://github.com/gabime/spdlog	https://github.com/gabime/spdlog?tab=License-1-ov-file#readme
spglib	https://spglib.readthedocs.io/	https://github.com/spglib/spglib/blob/develop/COPYING
sphinx	https://github.com/sphinx-doc/sphinx	https://github.com/sphinx-doc/sphinx/blob/5.x/LICENSE
sphinxcontrib-applehelp	https://github.com/sphinx-doc/sphinxcontrib-applehe	https://github.com/sphinx-doc/sphinxcontrib-applehe blob/master/LICENSE
sphinxcontrib-devhelp	https://github.com/sphinx-doc/sphinxcontrib-devhelp	https://github.com/sphinx-doc/sphinxcontrib-devhelp blob/master/LICENSE
sphinxcontrib-htmlhelp	https://github.com/sphinx-doc/sphinxcontrib-htmlhel	https://github.com/sphinx-doc/sphinxcontrib-htmlhel blob/master/LICENSE

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
sphinxcontrib-jsmath	https://github.com/sphinx-doc/sphinxcontrib-jsmath	https://github.com/sphinx-doc/sphinxcontrib-jsmath/blob/master/LICENSE
sphinxcontrib-qthelp	https://github.com/sphinx-doc/sphinxcontrib-qthelp	https://github.com/sphinx-doc/sphinxcontrib-qthelp/blob/master/LICENSE
sphinxcontrib-serializinghtml	https://github.com/sphinx-doc/sphinxcontrib-serializ	https://github.com/sphinx-doc/sphinxcontrib-serializ blob/master/LICENSE
SQLAlchemy	https://www.sqlalchemy.org	https://docs.sqlalchemy.org/en/14/copyright.html
SQLAlchemy	https://www.sqlalchemy.org/	https://github.com/sqlalchemy/sqlalchemy?tab=MIT-1-ov-file#readme
sqlite	https://sqlite.org/index.html	https://sqlite.org/copyright.html
sqlparse	https://github.com/andialbrecht/sqlparse	https://github.com/andialbrecht/sqlparse/blob/master/LICENSE
stack-data	http://github.com/alexmojaki/stack_data	https://github.com/alexmojaki/stack_data/blob/master/LICENSE.txt
stack_data	http://github.com/alexmojaki/stack_data	https://github.com/alexmojaki/stack_data?tab=MIT-1-ov-file#readme
starfile	https://github.com/alisterburt/starfile	https://github.com/teamtomo/starfile/blob/main/LICENSE
statsmodels	https://github.com/statsmodels/statsmodels	https://github.com/statsmodels/statsmodels/blob/main/LICENSE.txt
structlog	https://www.structlog.org/	https://github.com/hynek/structlog/blob/main/LICENSE

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
streamlit	https://github.com/streamlit/streamlit	https://github.com/streamlit/streamlit/blob/develop/LICENSE
svglib	https://github.com/deeplook/svglib	https://github.com/deeplook/svglib/blob/master/LICENSE.txt
sympy	https://sympy.org	https://github.com/sympy/sympy/blob/master/LICENSE
tables	https://www.pytables.org/	https://github.com/PyTables/PyTables/blob/master/LICENSE.txt
tabulate	https://github.com/astanin/python-tabulate	https://github.com/astanin/python-tabulate/blob/master/LICENSE
tbb	https://github.com/oneapi-src/oneTBB	https://github.com/oneapi-src/oneTBB/blob/master/LICENSE.txt
tenacity	https://github.com/jd/tenacity	https://github.com/jd/tenacity/blob/master/LICENSE
tensorboard	https://github.com/tensorflow/tensorboard	https://github.com/tensorflow/tensorboard/blob/master/LICENSE
tensorboard-data-server	https://github.com/tensorflow/tensorboard	https://github.com/tensorflow/tensorboard/blob/master/LICENSE
tensorboard-plugin-wit	https://github.com/pair-code/what-if-tool	https://github.com/PAIR-code/what-if-tool/blob/master/LICENSE
tensorboardX	https://github.com/lanpa/tensorboardX	https://github.com/lanpa/tensorboardX/blob/master/LICENSE
tensorflow	https://github.com/tensorflow/tensorflow	https://github.com/tensorflow/tensorflow/blob/master/LICENSE

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
tensorflow-estimator	https://github.com/tensorflow/estimator	https://github.com/tensorflow/estimator/blob/master/LICENSE
tensorflow-io-gcs-filesystem	https://github.com/tensorflow/io	https://github.com/tensorflow/io/blob/master/LICENSE
tensorflow-probability	https://github.com/tensorflow/probability	https://github.com/tensorflow/probability/blob/main/LICENSE
termcolor	https://github.com/hugovk/termcolor	https://github.com/hugovk/termcolor/blob/main/COPYING.txt
terminado	https://github.com/jupyter/terminado	https://github.com/jupyter/terminado/blob/main/LICENSE
testpath	https://github.com/jupyter/testpath	https://github.com/jupyter/testpath/blob/master/LICENSE
textangular	https://github.com/fraywing/textAngular	https://github.com/fraywing/textAngular
texttable	https://github.com/foutaise/texttable/	https://github.com/foutaise/texttable/?tab=MIT-1-ov-file#readme
tf_keras	https://github.com/keras-team/keras-io	https://github.com/keras-team/keras-io/blob/master/LICENSE
tf_keras	https://github.com/keras-team/keras-io	https://github.com/keras-team/keras-io/blob/master/LICENSE
threadpoolctl	https://github.com/joblib/threadpoolctl	https://github.com/joblib/threadpoolctl/blob/master/LICENSE
three	https://github.com/mrdoob/three.js	https://github.com/mrdoob/three.js/raw/master/LICENSE

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
tiffile	https://github.com/cgohlke/tiffile/	https://github.com/cgohlke/tiffile/blob/master/LICENSE
tinycss2	https://github.com/Kozea/tinycss2/	https://github.com/Kozea/tinycss2/blob/master/LICENSE
tinyxml2	https://github.com/leethomason/tinyxml2	https://github.com/leethomason/tinyxml2/blob/master/LICENSE.txt
tk	https://www.tcl.tk/	https://www.tcl.tk/software/tcltk/license.html
toml	https://github.com/toml-lang/toml	https://github.com/toml-lang/toml/blob/main/LICENSE
tomli	https://github.com/hukkin/tomli	https://github.com/hukkin/tomli/blob/master/LICENSE
toolz	https://github.com/pytoolz/toolz	https://github.com/pytoolz/toolz/blob/master/LICENSE.txt
torch	https://pytorch.org/	https://pypi.org/project/torch/#license
torch-geometric	https://pyg.org/	https://github.com/pyg-team/pytorch_geometric?tab=MIT-1-ov-file#readme
torcheval	https://github.com/pytorch/torcheval	https://github.com/pytorch/torcheval?tab=License-1-ov-file#readme
torchmetrics	https://github.com/Lightning-AI/torchmetrics	https://github.com/Lightning-AI/torchmetrics?tab=Apache-2.0-1-ov-file#readme
tornado	https://www.tornadoweb.org	https://github.com/tornadoweb/tornado/blob/master/LICENSE

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL		
tqdm	https://github.com/tqdm/tqdm	https://github.com/tqdm/tqdm/blob/master/LICENCE		
tracy	https://github.com/gear-genomics/tracy	https://github.com/gear-genomics/tracy/blob/main/LICENSE		
traitlets	https://github.com/ipython/traitlets	https://github.com/ipython/traitlets/blob/main/COPYING.md		
triton	https://github.com/openai/triton/	https://github.com/openai/triton/blob/main/LICENSE		
truststore	https://github.com/sethmlarson/trus	https://github.com/sethmlarson/truststore/blob/main/LICENSE		
truststore	https://github.com/sethmlarson/truststore	https://github.com/sethmlarson/truststore/blob/main/LICENSEtruststore	https://github.com/sethmlarson/truststore	https://github.com/sethmlarson/truststore/blob/main/LICENSE
ts-jest	https://github.com/kulshekhar/ts-jest	https://github.com/kulshekhar/ts-jest/raw/master/LICENSE.md		
ts-loader	https://github.com/TypeStrong/ts-loader	https://github.com/TypeStrong/ts-loader/raw/master/LICENSE		
twine	https://github.com/pypa/twine	https://github.com/pypa/twine/blob/main/LICENSE		
twinj uuid	https://github.com/twinj/uuid	https://github.com/twinj/uuid/blob/master/LICENSE		
types	https://github.com/babel/babel	https://github.com/babel/babel/raw/master/LICENSE		
types-python-dateutil	https://github.com/python/typeshed	https://github.com/python/typeshed?tab=License-1-ov-file#readme		
types-requests	https://pypi.org/project/types-requests/	https://github.com/python/typeshed/blob/main/LICENSE		

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
typescript	https://github.com/Microsoft/TypeScript	https://github.com/Microsoft/TypeScript/raw/master/LICENSE.txt
typing-inspection	https://github.com/pydantic/typing-inspection	https://github.com/pydantic/typing-inspection?tab=MIT-1-ov-file#readme
typing-utils	https://github.com/bojiang/typing_utils	https://github.com/bojiang/typing_utils?tab=Apache-2.0-1-ov-file#readme
typing_extensions	https://github.com/python/typing	https://github.com/python/typing/blob/master/LICENSE
typing_utils	https://github.com/bojiang/typing_utils	https://github.com/bojiang/typing_utils?tab=Apache-2.0-1-ov-file#readme
tzdata	https://github.com/python/tzdata	https://github.com/python/tzdata/blob/master/LICENSE
tzlocal	https://github.com/regebro/tzlocal	https://github.com/regebro/tzlocal/blob/master/LICENSE.txt
umf	https://github.com/oneapi-src/unified-memory-fram	https://github.com/oneapi-src/unified-memory-fram tab=License-1-ov-file#readme
umi-tools	https://github.com/CGATOxford/UMI-tools	https://github.com/CGATOxford/UMI-tools/blob/master/LICENSE
unicodedata2	https://github.com/fonttools/unicodedata2	https://github.com/fonttools/unicodedata2/blob/master/LICENSE
uri-template	https://gitlab.linss.com/open-source/python/uri-template	https://gitlab.linss.com/open-source/python/uri-template/-/blob/main/LICENSE
uritools	https://github.com/tkem/uritools/	https://github.com/tkem/uritools/blob/master/LICENSE

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL		
urllib3	https://urllib3.readthedocs.io/	https://github.com/urllib3/urllib3/blob/main/LICENSE.txt		
vine	https://github.com/celery/vine	https://github.com/celery/vine/blob/master/LICENSE		
vue	https://github.com/vuejs/vue	https://github.com/vuejs/vue/raw/master/LICENSE		
wadler_lindig	https://github.com/patrick-kidger/wadler_lindig	https://github.com/patrick-kidger/wadler_lindig?tab=Apache-2.0-1-ov-file#readme		
wadler_lindig	https://github.com/patrick-kidger/wadler_lindig	https://github.com/patrick-kidger/wadler_lindig?tab=Apache-2.0-1-ov-file#readmewcwidth	https://github.com/jquast/wcwidth	https://github.com/jquast/wcwidth/blob/master/LICENSE
wandb	https://github.com/wandb/wandb	https://github.com/wandb/wandb/blob/main/LICENSE		
webcolors	https://github.com/ubernostrum/webcolors	https://github.com/ubernostrum/webcolors?tab=BSD-3-Clause-1-ov-file#readme		
webencodings	https://github.com/gsnedders/python-webencodings	https://github.com/gsnedders/python-webencodings/blob/master/LICENSE		
websocket-client	https://github.com/websocket-client/websocket-client.git	https://github.com/websocket-client/websocket-client/blob/master/LICENSE		
Werkzeug	https://palletsprojects.com/p/werkzeug/	https://werkzeug.palletsprojects.com/en/2.1.x/license/		
westpa	http://github.com/westpa/westpa	https://github.com/westpa/westpa/blob/westpa2/LICENSE		
wheel	https://github.com/pypa/wheel	https://github.com/pypa/wheel/blob/main/LICENSE.txt		

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
widgetsnbextension	https://github.com/jupyter-widgets/ipywidgets#readme	https://github.com/jupyter-widgets/ipywidgets/blob/main/LICENSE
wrapt	https://github.com/GrahamDumpleton/wrapt	https://github.com/GrahamDumpleton/wrapt/blob/develop/LICENSE
wsutil	https://github.com/yhat/wsutil	https://github.com/yhat/wsutil/blob/master/LICENSE
x/lint	https://golang.org/x/lint	https://github.com/golang/lint/blob/master/LICENSE
x/mod	https://golang.org/x/mod/semver	https://cs.opensource.google/go/x/mod/+/master:LICENSE
x/net	https://golang.org/x/net	https://cs.opensource.google/go/x/net/+/master:LICENSE
x/oauth2	https://golang.org/x/oauth2	https://cs.opensource.google/go/x/oauth2/+/master:LICENSE
x/sys	https://golang.org/x/sys	https://cs.opensource.google/go/x/sys/+/master:LICENSE
x/text	https://golang.org/x/text	https://cs.opensource.google/go/x/text/+/master:LICENSE
x/tools	https://golang.org/x/tools	https://cs.opensource.google/go/x/tools/+/master:LICENSE
x/xerrors	https://golang.org/x/xerrors	https://cs.opensource.google/go/x/xerrors/+/master:LICENSE
xgboost	https://github.com/dmlc/xgboost	https://github.com/dmlc/xgboost/blob/master/LICENSE
xhtml2pdf	http://github.com/xhtml2pdf/xhtml2pdf	https://github.com/xhtml2pdf/xhtml2pdf/blob/master/LICENSE.txt

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
xlrd	https://github.com/python-excel/xlrd	https://github.com/python-excel/xlrd/blob/master/LICENSE
xlsxwriter	https://github.com/jmcnamara/XlsxWriter	https://github.com/jmcnamara/XlsxWriter/blob/main/LICENSE.txt
xmlsec	https://github.com/mehcode/python-xmlsec	https://github.com/mehcode/python-xmlsec/blob/master/LICENSE
xmltodict	https://github.com/martinblech/xmltodict	https://github.com/martinblech/xmltodict/blob/master/LICENSE
xorg-kbproto	https://gitlab.freedesktop.org/xorg/proto/kbproto	https://gitlab.freedesktop.org/xorg/proto/kbproto/-/blob/master/COPYING
xorg-libice	https://gitlab.freedesktop.org/xorg/lib/libice	https://gitlab.freedesktop.org/xorg/lib/libice/-/blob/master/COPYING
xorg-libsm	https://gitlab.freedesktop.org/xorg/lib/libsm	https://gitlab.freedesktop.org/xorg/lib/libsm/-/blob/master/COPYING
xorg-libx11	https://gitlab.freedesktop.org/xorg/lib/libx11	https://gitlab.freedesktop.org/xorg/lib/libx11/-/blob/master/COPYING
xorg-libxau	https://gitlab.freedesktop.org/xorg/lib/libxau	https://gitlab.freedesktop.org/xorg/lib/libxau/-/blob/master/COPYING
xorg-libxdmcp	https://gitlab.freedesktop.org/xorg/lib/libxdmcp	https://gitlab.freedesktop.org/xorg/lib/libxdmcp/-/blob/master/COPYING

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
xorg-libxext	https://gitlab.freedesktop.org/xorg/lib/libxext	https://gitlab.freedesktop.org/xorg/lib/libxext/-/blob/master/COPYING
xorg-libxrender	https://gitlab.freedesktop.org/xorg/lib/libxrender	https://gitlab.freedesktop.org/xorg/lib/libxrender/-/blob/master/COPYING
xorg-libxt	https://gitlab.freedesktop.org/xorg/lib/libxt	https://gitlab.freedesktop.org/xorg/lib/libxt/-/blob/master/COPYING
xorg-renderproto	https://gitlab.freedesktop.org/xorg/proto/renderproto	https://gitlab.freedesktop.org/xorg/proto/renderproto/-/blob/master/COPYING
xorg-xextproto	https://gitlab.freedesktop.org/xorg/proto/xextproto	https://gitlab.freedesktop.org/xorg/proto/xextproto/-/blob/master/COPYING
xorg-xproto	https://gitlab.freedesktop.org/xorg/proto/xproto	https://gitlab.freedesktop.org/xorg/proto/xproto/-/blob/master/COPYING
xxhash	https://github.com/cespare/xxhash/v2	https://github.com/cespare/xxhash/blob/main/LICENSE.txt
xz	https://github.com/ulikunitz/xz	https://github.com/ulikunitz/xz/blob/master/LICENSE
yaml	https://pyyaml.org/	https://github.com/yaml/pyyaml/blob/master/LICENSE
yaml-cpp	https://github.com/jbeder/yaml-cpp	https://github.com/jbeder/yaml-cpp/blob/master/LICENSE
yaml.v2	https://gopkg.in/yaml.v2	https://github.com/go-yaml/yaml/blob/v2.4.0/LICENSE
yaml.v3	https://gopkg.in/yaml.v3	https://github.com/go-yaml/yaml/blob/v3/LICENSE

continues on next page

Table 1 – continued from previous page

Name of Project	Website	License URL
yarl	https://github.com/aio-libs/yarl/	https://github.com/aio-libs/yarl/blob/master/LICENSE
yaspin	https://github.com/pavdmyt/yaspin	https://github.com/pavdmyt/yaspin/blob/master/LICENSE
yfiles	https://www.yworks.com/products/yfiles	commercial license
yml	https://pypi.org/project/yml/	N/A
zap	https://go.uber.org/zap	https://github.com/uber-go/zap/blob/master/LICENSE.txt
zipp	https://github.com/jaraco/zipp	https://github.com/jaraco/zipp/blob/main/LICENSE
zlib	https://zlib.net/	https://zlib.net/zlib_license.html
zstandard	https://github.com/indygreg/python-zstandard	https://github.com/indygreg/python-zstandard/blob/main/LICENSE
zstd	https://facebook.github.io/zstd/	https://github.com/facebook/zstd/blob/dev/LICENSE
click-option-group	https://github.com/click-contrib/click-option-group	https://github.com/click-contrib/click-option-group/blob/master/LICENSE

8.1 GCC

On the Linux platform, The OpenEye toolkits are linked with the GCC libraries using an eligible compilation process under the terms of the GCC RUNTIME LIBRARY EXCEPTION (see below).

The OpenEye toolkits are NOT open source software and are NOT governed by GPL or other open source licenses. Your right to use the toolkits are governed by the OpenEye license which is binding to You.

For certain Linux distributions, GCC library components are included as object files. Your rights to use GCC library components (libstdc++, libgcc, and libgomp) included with these distributions are governed by the GPLv3 license as follows:

The components libstdc++, libgcc, and libgomp are part of GCC.

- GCC is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.
- GCC is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the

implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with GCC (see below). If not, see Licenses GNU Project .

Per GPLv3, section 6d, the source for GCC is available for download from the following location: GNU. The GCC library components included here are from version gcc-4.9.3 and are built using the standard instructions at Installing GCC.

8.1.1 GCC RUNTIME LIBRARY EXCEPTION

```
Version 3.1, 31 March 2009

Copyright © 2009 Free Software Foundation, Inc. http://fsf.org/

Everyone is permitted to copy and distribute verbatim copies of this license document,␣
↪but changing it is not allowed.

This GCC Runtime Library Exception ("Exception") is an additional permission under␣
↪section 7 of the GNU General Public License,
version 3 ("GPLv3"). It applies to a given file (the "Runtime Library") that bears a␣
↪notice placed by the copyright holder
of the file stating that the file is governed by GPLv3 along with this Exception.

When you use GCC to compile a program, GCC may combine portions of certain GCC header␣
↪files and runtime libraries with
the compiled program. The purpose of this Exception is to allow compilation of non-GPL␣
↪(including proprietary) programs
to use, in this way, the header files and runtime libraries covered by this Exception.

0. Definitions.

A file is an "Independent Module" if it either requires the Runtime Library for␣
↪execution after a Compilation Process,
or makes use of an interface provided by the Runtime Library, but is not otherwise based␣
↪on the Runtime Library.

"GCC" means a version of the GNU Compiler Collection, with or without modifications,␣
↪governed by version 3 (or a specified
later version) of the GNU General Public License (GPL) with the option of using any␣
↪subsequent versions published by the FSF.

"GPL-compatible Software" is software whose conditions of propagation, modification and␣
↪use would permit combination with
GCC in accord with the license of GCC.

"Target Code" refers to output from any compiler for a real or virtual target processor␣
↪architecture, in executable
form or suitable for input to an assembler, loader, linker and/or execution phase.␣
↪Notwithstanding that, Target Code
does not include data in any format that is used as a compiler intermediate␣
↪representation, or used for producing a
compiler intermediate representation.
```

(continues on next page)

(continued from previous page)

```
The "Compilation Process" transforms code entirely represented in non-intermediate␣
↪languages designed for human-written
code, and/or in Java Virtual Machine byte code, into Target Code. Thus, for example, use␣
↪of source code generators and
preprocessors need not be considered part of the Compilation Process, since the␣
↪Compilation Process can be understood
as starting with the output of the generators or preprocessors.

A Compilation Process is "Eligible" if it is done using GCC, alone or with other GPL-
↪compatible software, or if it is
done without using any work based on GCC. For example, using non-GPL-compatible Software␣
↪to optimize any GCC intermediate
representations would not qualify as an Eligible Compilation Process.

1. Grant of Additional Permission.

You have permission to propagate a work of Target Code formed by combining the Runtime␣
↪Library with Independent Modules,
even if such propagation would otherwise violate the terms of GPLv3, provided that all␣
↪Target Code was generated by
Eligible Compilation Processes. You may then convey such a combination under terms of␣
↪your choice, consistent with the
licensing of the Independent Modules.

2. No Weakening of GCC Copyleft.

The availability of this Exception does not imply any general presumption that third-
↪party software is unaffected by
the copyleft requirements of the license of GCC.
```

See also

- http://www.gnu.org/licenses/gcc-exception.html

8.1.2 GNU GENERAL PUBLIC LICENSE

```
                    GNU GENERAL PUBLIC LICENSE
                       Version 3, 29 June 2007

 Copyright (C) 2007 Free Software Foundation, Inc. <http://fsf.org/>
 Everyone is permitted to copy and distribute verbatim copies
 of this license document, but changing it is not allowed.

                            Preamble

 The GNU General Public License is a free, copyleft license for
 software and other kinds of works.

   The licenses for most software and other practical works are designed
```

(continues on next page)

(continued from previous page)

```
to take away your freedom to share and change the works.  By contrast,
the GNU General Public License is intended to guarantee your freedom to
share and change all versions of a program--to make sure it remains free
software for all its users.  We, the Free Software Foundation, use the
GNU General Public License for most of our software; it applies also to
any other work released this way by its authors.  You can apply it to
your programs, too.

  When we speak of free software, we are referring to freedom, not
price.  Our General Public Licenses are designed to make sure that you
have the freedom to distribute copies of free software (and charge for
them if you wish), that you receive source code or can get it if you
want it, that you can change the software or use pieces of it in new
free programs, and that you know you can do these things.

  To protect your rights, we need to prevent others from denying you
these rights or asking you to surrender the rights.  Therefore, you have
certain responsibilities if you distribute copies of the software, or if
you modify it: responsibilities to respect the freedom of others.

  For example, if you distribute copies of such a program, whether
gratis or for a fee, you must pass on to the recipients the same
freedoms that you received.  You must make sure that they, too, receive
or can get the source code.  And you must show them these terms so they
know their rights.

  Developers that use the GNU GPL protect your rights with two steps:
(1) assert copyright on the software, and (2) offer you this License
giving you legal permission to copy, distribute and/or modify it.

  For the developers' and authors' protection, the GPL clearly explains
that there is no warranty for this free software.  For both users' and
authors' sake, the GPL requires that modified versions be marked as
changed, so that their problems will not be attributed erroneously to
authors of previous versions.

  Some devices are designed to deny users access to install or run
modified versions of the software inside them, although the manufacturer
can do so.  This is fundamentally incompatible with the aim of
protecting users' freedom to change the software.  The systematic
pattern of such abuse occurs in the area of products for individuals to
use, which is precisely where it is most unacceptable.  Therefore, we
have designed this version of the GPL to prohibit the practice for those
products.  If such problems arise substantially in other domains, we
stand ready to extend this provision to those domains in future versions
of the GPL, as needed to protect the freedom of users.

  Finally, every program is threatened constantly by software patents.
States should not allow patents to restrict development and use of
software on general-purpose computers, but in those that do, we wish to
avoid the special danger that patents applied to a free program could
make it effectively proprietary.  To prevent this, the GPL assures that
```

(continues on next page)

(continued from previous page)

```
patents cannot be used to render the program non-free.

  The precise terms and conditions for copying, distribution and
modification follow.

                    TERMS AND CONDITIONS

  0. Definitions.

  "This License" refers to version 3 of the GNU General Public License.

  "Copyright" also means copyright-like laws that apply to other kinds of
works, such as semiconductor masks.

  "The Program" refers to any copyrightable work licensed under this
License.  Each licensee is addressed as "you".  "Licensees" and
"recipients" may be individuals or organizations.

  To "modify" a work means to copy from or adapt all or part of the work
in a fashion requiring copyright permission, other than the making of an
exact copy.  The resulting work is called a "modified version" of the
earlier work or a work "based on" the earlier work.

  A "covered work" means either the unmodified Program or a work based
on the Program.

  To "propagate" a work means to do anything with it that, without
permission, would make you directly or secondarily liable for
infringement under applicable copyright law, except executing it on a
computer or modifying a private copy.  Propagation includes copying,
distribution (with or without modification), making available to the
public, and in some countries other activities as well.

  To "convey" a work means any kind of propagation that enables other
parties to make or receive copies.  Mere interaction with a user through
a computer network, with no transfer of a copy, is not conveying.

  An interactive user interface displays "Appropriate Legal Notices"
to the extent that it includes a convenient and prominently visible
feature that (1) displays an appropriate copyright notice, and (2)
tells the user that there is no warranty for the work (except to the
extent that warranties are provided), that licensees may convey the
work under this License, and how to view a copy of this License.  If
the interface presents a list of user commands or options, such as a
menu, a prominent item in the list meets this criterion.

  1. Source Code.

  The "source code" for a work means the preferred form of the work
for making modifications to it.  "Object code" means any non-source
form of a work.
```

(continues on next page)

(continued from previous page)

```
  A "Standard Interface" means an interface that either is an official
standard defined by a recognized standards body, or, in the case of
interfaces specified for a particular programming language, one that
is widely used among developers working in that language.

  The "System Libraries" of an executable work include anything, other
than the work as a whole, that (a) is included in the normal form of
packaging a Major Component, but which is not part of that Major
Component, and (b) serves only to enable use of the work with that
Major Component, or to implement a Standard Interface for which an
implementation is available to the public in source code form.  A
"Major Component", in this context, means a major essential component
(kernel, window system, and so on) of the specific operating system
(if any) on which the executable work runs, or a compiler used to
produce the work, or an object code interpreter used to run it.

  The "Corresponding Source" for a work in object code form means all
the source code needed to generate, install, and (for an executable
work) run the object code and to modify the work, including scripts to
control those activities.  However, it does not include the work's
System Libraries, or general-purpose tools or generally available free
programs which are used unmodified in performing those activities but
which are not part of the work.  For example, Corresponding Source
includes interface definition files associated with source files for
the work, and the source code for shared libraries and dynamically
linked subprograms that the work is specifically designed to require,
such as by intimate data communication or control flow between those
subprograms and other parts of the work.

  The Corresponding Source need not include anything that users
can regenerate automatically from other parts of the Corresponding
Source.

  The Corresponding Source for a work in source code form is that
same work.

  2. Basic Permissions.

  All rights granted under this License are granted for the term of
copyright on the Program, and are irrevocable provided the stated
conditions are met.  This License explicitly affirms your unlimited
permission to run the unmodified Program.  The output from running a
covered work is covered by this License only if the output, given its
content, constitutes a covered work.  This License acknowledges your
rights of fair use or other equivalent, as provided by copyright law.

  You may make, run and propagate covered works that you do not
convey, without conditions so long as your license otherwise remains
in force.  You may convey covered works to others for the sole purpose
of having them make modifications exclusively for you, or provide you
with facilities for running those works, provided that you comply with
the terms of this License in conveying all material for which you do
```

(continues on next page)

(continued from previous page)

```
not control copyright.  Those thus making or running the covered works
for you must do so exclusively on your behalf, under your direction
and control, on terms that prohibit them from making any copies of
your copyrighted material outside their relationship with you.

  Conveying under any other circumstances is permitted solely under
the conditions stated below.  Sublicensing is not allowed; section 10
makes it unnecessary.

  3. Protecting Users' Legal Rights From Anti-Circumvention Law.

  No covered work shall be deemed part of an effective technological
measure under any applicable law fulfilling obligations under article
11 of the WIPO copyright treaty adopted on 20 December 1996, or
similar laws prohibiting or restricting circumvention of such
measures.

  When you convey a covered work, you waive any legal power to forbid
circumvention of technological measures to the extent such circumvention
is effected by exercising rights under this License with respect to
the covered work, and you disclaim any intention to limit operation or
modification of the work as a means of enforcing, against the work's
users, your or third parties' legal rights to forbid circumvention of
technological measures.

  4. Conveying Verbatim Copies.

  You may convey verbatim copies of the Program's source code as you
receive it, in any medium, provided that you conspicuously and
appropriately publish on each copy an appropriate copyright notice;
keep intact all notices stating that this License and any
non-permissive terms added in accord with section 7 apply to the code;
keep intact all notices of the absence of any warranty; and give all
recipients a copy of this License along with the Program.


  You may charge any price or no price for each copy that you convey,
and you may offer support or warranty protection for a fee.

  5. Conveying Modified Source Versions.

  You may convey a work based on the Program, or the modifications to
produce it from the Program, in the form of source code under the
terms of section 4, provided that you also meet all of these conditions:

   a) The work must carry prominent notices stating that you modified
   it, and giving a relevant date.

   b) The work must carry prominent notices stating that it is
   released under this License and any conditions added under section
   7.  This requirement modifies the requirement in section 4 to
   "keep intact all notices".
```

(continues on next page)

(continued from previous page)

```
    c) You must license the entire work, as a whole, under this
    License to anyone who comes into possession of a copy.  This
    License will therefore apply, along with any applicable section 7
    additional terms, to the whole of the work, and all its parts,
    regardless of how they are packaged.  This License gives no
    permission to license the work in any other way, but it does not
    invalidate such permission if you have separately received it.

    d) If the work has interactive user interfaces, each must display
    Appropriate Legal Notices; however, if the Program has interactive
    interfaces that do not display Appropriate Legal Notices, your
    work need not make them do so.

  A compilation of a covered work with other separate and independent
works, which are not by their nature extensions of the covered work,
and which are not combined with it such as to form a larger program,
in or on a volume of a storage or distribution medium, is called an
"aggregate" if the compilation and its resulting copyright are not
used to limit the access or legal rights of the compilation's users
beyond what the individual works permit.  Inclusion of a covered work
in an aggregate does not cause this License to apply to the other
parts of the aggregate.

  6. Conveying Non-Source Forms.

  You may convey a covered work in object code form under the terms
of sections 4 and 5, provided that you also convey the
machine-readable Corresponding Source under the terms of this License,
in one of these ways:

  a) Convey the object code in, or embodied in, a physical product
  (including a physical distribution medium), accompanied by the
  Corresponding Source fixed on a durable physical medium
  customarily used for software interchange.

  b) Convey the object code in, or embodied in, a physical product
  (including a physical distribution medium), accompanied by a
  written offer, valid for at least three years and valid for as
  long as you offer spare parts or customer support for that product
  model, to give anyone who possesses the object code either (1) a
  copy of the Corresponding Source for all the software in the
  product that is covered by this License, on a durable physical
  medium customarily used for software interchange, for a price no
  more than your reasonable cost of physically performing this
  conveying of source, or (2) access to copy the
  Corresponding Source from a network server at no charge.

  c) Convey individual copies of the object code with a copy of the
  written offer to provide the Corresponding Source.  This
  alternative is allowed only occasionally and noncommercially, and
  only if you received the object code with such an offer, in accord
```

(continues on next page)

(continued from previous page)

```
   with subsection 6b.

   d) Convey the object code by offering access from a designated
   place (gratis or for a charge), and offer equivalent access to the
   Corresponding Source in the same way through the same place at no
   further charge.  You need not require recipients to copy the
   Corresponding Source along with the object code.  If the place to
   copy the object code is a network server, the Corresponding Source
   may be on a different server (operated by you or a third party)
   that supports equivalent copying facilities, provided you maintain
   clear directions next to the object code saying where to find the
   Corresponding Source.  Regardless of what server hosts the
   Corresponding Source, you remain obligated to ensure that it is
   available for as long as needed to satisfy these requirements.

   e) Convey the object code using peer-to-peer transmission, provided
   you inform other peers where the object code and Corresponding
   Source of the work are being offered to the general public at no
   charge under subsection 6d.

  A separable portion of the object code, whose source code is excluded
from the Corresponding Source as a System Library, need not be
included in conveying the object code work.

  A "User Product" is either (1) a "consumer product", which means any
tangible personal property which is normally used for personal, family,
or household purposes, or (2) anything designed or sold for incorporation
into a dwelling.  In determining whether a product is a consumer product,
doubtful cases shall be resolved in favor of coverage.  For a particular
product received by a particular user, "normally used" refers to a
typical or common use of that class of product, regardless of the status
of the particular user or of the way in which the particular user
actually uses, or expects or is expected to use, the product.  A product
is a consumer product regardless of whether the product has substantial
commercial, industrial or non-consumer uses, unless such uses represent
the only significant mode of use of the product.

  "Installation Information" for a User Product means any methods,
procedures, authorization keys, or other information required to install
and execute modified versions of a covered work in that User Product from
a modified version of its Corresponding Source.  The information must
suffice to ensure that the continued functioning of the modified object
code is in no case prevented or interfered with solely because
modification has been made.

  If you convey an object code work under this section in, or with, or
specifically for use in, a User Product, and the conveying occurs as
part of a transaction in which the right of possession and use of the
User Product is transferred to the recipient in perpetuity or for a
fixed term (regardless of how the transaction is characterized), the
Corresponding Source conveyed under this section must be accompanied
by the Installation Information.  But this requirement does not apply
```

(continues on next page)

(continued from previous page)

```
if neither you nor any third party retains the ability to install
modified object code on the User Product (for example, the work has
been installed in ROM).

  The requirement to provide Installation Information does not include a
requirement to continue to provide support service, warranty, or updates
for a work that has been modified or installed by the recipient, or for
the User Product in which it has been modified or installed.  Access to a
network may be denied when the modification itself materially and
adversely affects the operation of the network or violates the rules and
protocols for communication across the network.

  Corresponding Source conveyed, and Installation Information provided,
in accord with this section must be in a format that is publicly
documented (and with an implementation available to the public in
source code form), and must require no special password or key for
unpacking, reading or copying.

  7. Additional Terms.

  "Additional permissions" are terms that supplement the terms of this
License by making exceptions from one or more of its conditions.
Additional permissions that are applicable to the entire Program shall
be treated as though they were included in this License, to the extent
that they are valid under applicable law.  If additional permissions
apply only to part of the Program, that part may be used separately
under those permissions, but the entire Program remains governed by
this License without regard to the additional permissions.

  When you convey a copy of a covered work, you may at your option
remove any additional permissions from that copy, or from any part of
it.  (Additional permissions may be written to require their own
removal in certain cases when you modify the work.)  You may place
additional permissions on material, added by you to a covered work,
for which you have or can give appropriate copyright permission.

  Notwithstanding any other provision of this License, for material you
add to a covered work, you may (if authorized by the copyright holders of
that material) supplement the terms of this License with terms:

    a) Disclaiming warranty or limiting liability differently from the
    terms of sections 15 and 16 of this License; or

    b) Requiring preservation of specified reasonable legal notices or
    author attributions in that material or in the Appropriate Legal
    Notices displayed by works containing it; or

    c) Prohibiting misrepresentation of the origin of that material, or
    requiring that modified versions of such material be marked in
    reasonable ways as different from the original version; or

    d) Limiting the use for publicity purposes of names of licensors or
```

(continues on next page)

(continued from previous page)

```
   authors of the material; or

   e) Declining to grant rights under trademark law for use of some
   trade names, trademarks, or service marks; or

   f) Requiring indemnification of licensors and authors of that
   material by anyone who conveys the material (or modified versions of
   it) with contractual assumptions of liability to the recipient, for
   any liability that these contractual assumptions directly impose on
   those licensors and authors.

  All other non-permissive additional terms are considered "further
restrictions" within the meaning of section 10.  If the Program as you
received it, or any part of it, contains a notice stating that it is
governed by this License along with a term that is a further
restriction, you may remove that term.  If a license document contains
a further restriction but permits relicensing or conveying under this
License, you may add to a covered work material governed by the terms
of that license document, provided that the further restriction does
not survive such relicensing or conveying.

  If you add terms to a covered work in accord with this section, you
must place, in the relevant source files, a statement of the
additional terms that apply to those files, or a notice indicating
where to find the applicable terms.

  Additional terms, permissive or non-permissive, may be stated in the
form of a separately written license, or stated as exceptions;
the above requirements apply either way.

  8. Termination.

  You may not propagate or modify a covered work except as expressly
provided under this License.  Any attempt otherwise to propagate or
modify it is void, and will automatically terminate your rights under
this License (including any patent licenses granted under the third
paragraph of section 11).

  However, if you cease all violation of this License, then your
license from a particular copyright holder is reinstated (a)
provisionally, unless and until the copyright holder explicitly and
finally terminates your license, and (b) permanently, if the copyright
holder fails to notify you of the violation by some reasonable means
prior to 60 days after the cessation.

  Moreover, your license from a particular copyright holder is
reinstated permanently if the copyright holder notifies you of the
violation by some reasonable means, this is the first time you have
received notice of violation of this License (for any work) from that
copyright holder, and you cure the violation prior to 30 days after
your receipt of the notice.
```

(continues on next page)

(continued from previous page)

```
  Termination of your rights under this section does not terminate the
licenses of parties who have received copies or rights from you under
this License.  If your rights have been terminated and not permanently
reinstated, you do not qualify to receive new licenses for the same
material under section 10.

  9. Acceptance Not Required for Having Copies.

  You are not required to accept this License in order to receive or
run a copy of the Program.  Ancillary propagation of a covered work
occurring solely as a consequence of using peer-to-peer transmission
to receive a copy likewise does not require acceptance.  However,
nothing other than this License grants you permission to propagate or
modify any covered work.  These actions infringe copyright if you do
not accept this License.  Therefore, by modifying or propagating a
covered work, you indicate your acceptance of this License to do so.

  10. Automatic Licensing of Downstream Recipients.

  Each time you convey a covered work, the recipient automatically
receives a license from the original licensors, to run, modify and
propagate that work, subject to this License.  You are not responsible
for enforcing compliance by third parties with this License.

  An "entity transaction" is a transaction transferring control of an
organization, or substantially all assets of one, or subdividing an
organization, or merging organizations.  If propagation of a covered
work results from an entity transaction, each party to that
transaction who receives a copy of the work also receives whatever
licenses to the work the party's predecessor in interest had or could
give under the previous paragraph, plus a right to possession of the
Corresponding Source of the work from the predecessor in interest, if
the predecessor has it or can get it with reasonable efforts.

  You may not impose any further restrictions on the exercise of the
rights granted or affirmed under this License.  For example, you may
not impose a license fee, royalty, or other charge for exercise of
rights granted under this License, and you may not initiate litigation
(including a cross-claim or counterclaim in a lawsuit) alleging that
any patent claim is infringed by making, using, selling, offering for
sale, or importing the Program or any portion of it.

  11. Patents.

  A "contributor" is a copyright holder who authorizes use under this
License of the Program or a work on which the Program is based.  The
work thus licensed is called the contributor's "contributor version".

  A contributor's "essential patent claims" are all patent claims
owned or controlled by the contributor, whether already acquired or
hereafter acquired, that would be infringed by some manner, permitted
by this License, of making, using, or selling its contributor version,
```

(continues on next page)

```
but do not include claims that would be infringed only as a
consequence of further modification of the contributor version.  For
purposes of this definition, "control" includes the right to grant
patent sublicenses in a manner consistent with the requirements of
this License.

  Each contributor grants you a non-exclusive, worldwide, royalty-free
patent license under the contributor's essential patent claims, to
make, use, sell, offer for sale, import and otherwise run, modify and
propagate the contents of its contributor version.

  In the following three paragraphs, a "patent license" is any express
agreement or commitment, however denominated, not to enforce a patent
(such as an express permission to practice a patent or covenant not to
sue for patent infringement).  To "grant" such a patent license to a
party means to make such an agreement or commitment not to enforce a
patent against the party.

  If you convey a covered work, knowingly relying on a patent license,
and the Corresponding Source of the work is not available for anyone
to copy, free of charge and under the terms of this License, through a
publicly available network server or other readily accessible means,
then you must either (1) cause the Corresponding Source to be so
available, or (2) arrange to deprive yourself of the benefit of the
patent license for this particular work, or (3) arrange, in a manner
consistent with the requirements of this License, to extend the patent
license to downstream recipients.  "Knowingly relying" means you have
actual knowledge that, but for the patent license, your conveying the
covered work in a country, or your recipient's use of the covered work
in a country, would infringe one or more identifiable patents in that
country that you have reason to believe are valid.

  If, pursuant to or in connection with a single transaction or
arrangement, you convey, or propagate by procuring conveyance of, a
covered work, and grant a patent license to some of the parties
receiving the covered work authorizing them to use, propagate, modify
or convey a specific copy of the covered work, then the patent license
you grant is automatically extended to all recipients of the covered
work and works based on it.

  A patent license is "discriminatory" if it does not include within
the scope of its coverage, prohibits the exercise of, or is
conditioned on the non-exercise of one or more of the rights that are
specifically granted under this License.  You may not convey a covered
work if you are a party to an arrangement with a third party that is
in the business of distributing software, under which you make payment
to the third party based on the extent of your activity of conveying
the work, and under which the third party grants, to any of the
parties who would receive the covered work from you, a discriminatory
patent license (a) in connection with copies of the covered work
conveyed by you (or copies made from those copies), or (b) primarily
for and in connection with specific products or compilations that
contain the covered work, unless you entered into that arrangement,
```

(continues on next page)

(continued from previous page)

```
or that patent license was granted, prior to 28 March 2007.

  Nothing in this License shall be construed as excluding or limiting
any implied license or other defenses to infringement that may
otherwise be available to you under applicable patent law.

  12. No Surrender of Others' Freedom.

  If conditions are imposed on you (whether by court order, agreement or
otherwise) that contradict the conditions of this License, they do not
excuse you from the conditions of this License.  If you cannot convey a
covered work so as to satisfy simultaneously your obligations under this
License and any other pertinent obligations, then as a consequence you may
not convey it at all.  For example, if you agree to terms that obligate you
to collect a royalty for further conveying from those to whom you convey
the Program, the only way you could satisfy both those terms and this
License would be to refrain entirely from conveying the Program.

  13. Use with the GNU Affero General Public License.

  Notwithstanding any other provision of this License, you have
permission to link or combine any covered work with a work licensed
under version 3 of the GNU Affero General Public License into a single
combined work, and to convey the resulting work.  The terms of this
License will continue to apply to the part which is the covered work,
but the special requirements of the GNU Affero General Public License,
section 13, concerning interaction through a network will apply to the
combination as such.

  14. Revised Versions of this License.

  The Free Software Foundation may publish revised and/or new versions of
the GNU General Public License from time to time.  Such new versions will
be similar in spirit to the present version, but may differ in detail to
address new problems or concerns.

  Each version is given a distinguishing version number.  If the
Program specifies that a certain numbered version of the GNU General
Public License "or any later version" applies to it, you have the
option of following the terms and conditions either of that numbered
version or of any later version published by the Free Software
Foundation.  If the Program does not specify a version number of the
GNU General Public License, you may choose any version ever published
by the Free Software Foundation.

  If the Program specifies that a proxy can decide which future
versions of the GNU General Public License can be used, that proxy's
public statement of acceptance of a version permanently authorizes you
to choose that version for the Program.

  Later license versions may give you additional or different
permissions.  However, no additional obligations are imposed on any
```

(continues on next page)

(continued from previous page)

```
author or copyright holder as a result of your choosing to follow a
later version.

  15. Disclaimer of Warranty.

  THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY
APPLICABLE LAW.  EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT
HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY
OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO,
THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
PURPOSE.  THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM
IS WITH YOU.  SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF
ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

  16. Limitation of Liability.

  IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING
WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MODIFIES AND/OR CONVEYS
THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY
GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE
USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF
DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD
PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS),
EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF
SUCH DAMAGES.

  17. Interpretation of Sections 15 and 16.

  If the disclaimer of warranty and limitation of liability provided
above cannot be given local legal effect according to their terms,
reviewing courts shall apply local law that most closely approximates
an absolute waiver of all civil liability in connection with the
Program, unless a warranty or assumption of liability accompanies a
copy of the Program in return for a fee.

                     END OF TERMS AND CONDITIONS

            How to Apply These Terms to Your New Programs

  If you develop a new program, and you want it to be of the greatest
possible use to the public, the best way to achieve this is to make it
free software which everyone can redistribute and change under these terms.

  To do so, attach the following notices to the program.  It is safest
to attach them to the start of each source file to most effectively
state the exclusion of warranty; and each file should have at least
the "copyright" line and a pointer to where the full notice is found.

    <one line to give the program's name and a brief idea of what it does.>
    Copyright (C) <year>  <name of author>

    This program is free software: you can redistribute it and/or modify
```

(continues on next page)

(continued from previous page)

```
    it under the terms of the GNU General Public License as published by
    the Free Software Foundation, either version 3 of the License, or
    (at your option) any later version.

    This program is distributed in the hope that it will be useful,
    but WITHOUT ANY WARRANTY; without even the implied warranty of
    MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
    GNU General Public License for more details.

    You should have received a copy of the GNU General Public License
    along with this program.  If not, see <http://www.gnu.org/licenses/>.

Also add information on how to contact you by electronic and paper mail.

  If the program does terminal interaction, make it output a short
notice like this when it starts in an interactive mode:

    <program>  Copyright (C) <year>  <name of author>
    This program comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
    This is free software, and you are welcome to redistribute it
    under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate
parts of the General Public License.  Of course, your program's commands
might be different; for a GUI interface, you would use an "about box".

  You should also get your employer (if you work as a programmer) or school,
if any, to sign a "copyright disclaimer" for the program, if necessary.
For more information on this, and how to apply and follow the GNU GPL, see
<http://www.gnu.org/licenses/>.

  The GNU General Public License does not permit incorporating your program
into proprietary programs.  If your program is a subroutine library, you
may consider it more useful to permit linking proprietary applications with
the library.  If this is what you want to do, use the GNU Lesser General
Public License instead of this License.  But first, please read
<http://www.gnu.org/philosophy/why-not-lgpl.html>.
```

See also

- http://www.gnu.org/licenses/gpl.txt

INDEX

A

AdaptGrad
    OEMolPotential::OEInterAdaptor, 55
    OEMolPotential::OEQuatAdaptor, 86
    OEMolPotential::OESubsetAdaptor, 92
    OEMolPotential::OETorAdaptor, 94
    OEMolPotential::OETorQuatAdaptor, 96
AdaptHessian
    OEMolPotential::OESubsetAdaptor, 92
AdaptHostCoords
    OEFF::OEComplexFF, 152
    OEMolPotential::OEInterMolFunc1, 57
AdaptVar
    OEOpt::OEAdaptor, 12
AddBendData
    OEMolSmirnoff::OESmirnoffParams, 126
AddData
    OEMolSmirnoff::OESmirnoffTorsionData, 132
AddImproperData
    OEMolSmirnoff::OESmirnoffParams, 126
AddMolFunc
    OEMolPotential::OEGenericFF, 48
    OEMolPotential::OEGenericFF2, 50
AddStretchData
    OEMolSmirnoff::OESmirnoffParams, 126
AddTorsionData
    OEMolSmirnoff::OESmirnoffParams, 126
AddVdWData
    OEMolSmirnoff::OESmirnoffParams, 126
Assignment operator
    OEMolPotential::OEBendPotential, 35
    OEMolPotential::OECoulombPotential, 37
    OEMolPotential::OEFFParams, 38
    OEMolPotential::OEFFPotential, 44
    OEMolPotential::OEInteractParams, 53
    OEMolPotential::OEInterCoulombPotential, 56
    OEMolPotential::OEInterNonBondPotBase, 60
    OEMolPotential::OEInterNonBondPotential, 61
    OEMolPotential::OEInterVdwPotential, 62
    OEMolPotential::OENonBondIntcsOptions, 79
    OEMolPotential::OENonBondPotBase, 77
    OEMolPotential::OENonBondPotential, 79
    OEMolPotential::OEOutOfPlanePotential, 84
    OEMolPotential::OEStretchPotential, 90
    OEMolPotential::OETorsionPotential, 98
    OEMolPotential::OEVdwPotential, 101

C

Clear
    OEMolPotential::OEInteractParams, 53
    OEMolPotential::OEMolFunc, 68
ClearData
    OEMolSmirnoff::OESmirnoffTorsionData, 133
ClearPredicates
    OEMolPotential::OEMolFunc, 68
Constructors
    OEConstr::OEDihedralQuadPenalty, 109
    OEConstr::OEHarmonicPotential, 110
    OEFF::OEComplexEnergies, 150
    OEFF::OEComplexFF, 152
    OEFF::OEComplexFFParameter, 155
    OEFF::OEFF14SBParsley, 140
    OEFF::OEFF14SBParsleyComplex, 141
    OEFF::OEFF14SBParsleyParams, 143
    OEFF::OEFF14SBSage, 144
    OEFF::OEFF14SBSageComplex, 144
    OEFF::OEFF14SBSageParams, 146
    OEFF::OEFF14SBSmirnoff, 147
    OEFF::OEFF14SBSmirnoffComplex, 148
    OEFF::OEFF14SBSmirnoffParams, 149
    OEFF::OEForceFieldParameter, 156
    OEFF::OEMMFFAmber, 159
    OEFF::OEMMFFAmberComplex, 160
    OEFF::OEMMFFAmberOptions, 161
    OEFF::OEMMFFComplex, 163
    OEFF::OEMMFFIEFF, 158
    OEFF::OEMMFFIEFFOptions, 164
    OEFF::OEMMFFSheffield, 166
    OEFF::OEMMFFSheffieldOptions, 167
    OEFF::OESmirnoffComplex, 168
    OEMolMMFF::OEMMFF, 113
    OEMolMMFF::OEMMFF94sParams, 114

OEMolMMFF::OEMMFFOutOfPlane, 115
OEMolMMFF::OEMMFFParams, 117
OEMolMMFF::OEMMFFStretchBend, 118
OEMolPotential::OEBendParams, 34
OEMolPotential::OEBendPotential, 35
OEMolPotential::OECoulombPotential, 36
OEMolPotential::OEFFParams, 38
OEMolPotential::OEFFPotential, 44
OEMolPotential::OEGenericFF, 48
OEMolPotential::OEGenericFF2, 50
OEMolPotential::OEInteraction, 51
OEMolPotential::OEInteractParams, 52
OEMolPotential::OEInterAdaptor, 55
OEMolPotential::OEInterCoulombPotential, 56
OEMolPotential::OEInterNonBondPotBase, 59
OEMolPotential::OEInterNonBondPotential, 61
OEMolPotential::OEInterVdwPotential, 62
OEMolPotential::OENonBondIntcsOptions, 79
OEMolPotential::OENonBondPotBase, 77
OEMolPotential::OENonBondPotential, 79
OEMolPotential::OENumericMolFunc2, 81
OEMolPotential::OEOutOfPlaneParams, 82
OEMolPotential::OEOutOfPlanePotential, 84
OEMolPotential::OEQuatAdaptor, 85
OEMolPotential::OEScaledMolFunc, 87
OEMolPotential::OEStretchParams, 88
OEMolPotential::OEStretchPotential, 90
OEMolPotential::OESubsetAdaptor, 92
OEMolPotential::OETorAdaptor, 94
OEMolPotential::OETorQuatAdaptor, 95
OEMolPotential::OETorsionParams, 97
OEMolPotential::OETorsionPotential, 98
OEMolPotential::OEVdwParams, 99
OEMolPotential::OEVdwPotential, 101
OEMolSmirnoff::OEParsley, 120
OEMolSmirnoff::OEParsleyParams, 121
OEMolSmirnoff::OESage, 122
OEMolSmirnoff::OESageParams, 123
OEMolSmirnoff::OESmirnoff, 124
OEMolSmirnoff::OESmirnoffBendData, 131
OEMolSmirnoff::OESmirnoffImproperData, 133
OEMolSmirnoff::OESmirnoffParams, 126
OEMolSmirnoff::OESmirnoffStretchData, 130
OEMolSmirnoff::OESmirnoffTorsionData, 132
OEMolSmirnoff::OESmirnoffVdWData, 134
OEOpt::OEBFGSOpt, 13
OEOpt::OECGLineMinimize, 14
OEOpt::OEConjGradOpt, 16
OEOpt::OEDFPOpt, 16
OEOpt::OEFComponent, 17
OEOpt::OEFletcherReevesOpt, 17
OEOpt::OEHestenesStiefelOpt, 25
OEOpt::OEMaxStepLineMinimize, 27
OEOpt::OENewtonOpt, 33
OEOpt::OENRLineSearch, 27
OEOpt::OEQNLineMinimize, 31
OEOpt::OERibierePolakOpt, 32
OEOpt::OESteepestDescentOpt, 32
OESheff::OESheffield, 137
OESheff::OESheffieldOptions, 138
CreateCopy
OEFF::OEMMFFAmberOptions, 161
OEMolPotential::OEFFParams, 38
OEMolPotential::OENonBondIntcsOptions, 80
OEOpt::OELineMinimize, 26

G

Get
OEMolPotential::OEInteractParams, 53
Get_dc
OEConstr::OEHarmonicPotential, 111
Get_kc
OEConstr::OEHarmonicPotential, 111
GetA
OESheff::OESheffieldOptions, 138
GetAmberTerms
OEFF::OEMMFFAmberOptions, 161
GetAtom1
OEMolPotential::OEVdwParams, 100
GetAtom2
OEMolPotential::OEVdwParams, 100
GetAtomIndex1
OEMolPotential::OEBendParams, 34
OEMolPotential::OEOutOfPlaneParams, 83
OEMolPotential::OEStretchParams, 89
OEMolPotential::OETorsionParams, 97
OEMolPotential::OEVdwParams, 99
GetAtomIndex2
OEMolPotential::OEBendParams, 34
OEMolPotential::OEOutOfPlaneParams, 83
OEMolPotential::OEStretchParams, 89
OEMolPotential::OETorsionParams, 97
OEMolPotential::OEVdwParams, 100
GetAtomIndex3
OEMolPotential::OEBendParams, 34
OEMolPotential::OEOutOfPlaneParams, 83
OEMolPotential::OETorsionParams, 97
GetAtomIndex4
OEMolPotential::OEOutOfPlaneParams, 83
OEMolPotential::OETorsionParams, 97
GetAtoms
OEMolPotential::OEInteraction, 51
GetAtomTypeIndex
OEMolPotential::OEFFParams, 38
GetAtomTypeName

OEMolPotential::OEFFParams, 38
GetB
OESheff::OESheffieldOptions, 138
GetBendData
OEMolSmirnoff::OESmirnoffParams, 127
GetBendParams
OEMolPotential::OEFFParams, 41
GetC
OESheff::OESheffieldOptions, 138
GetComplexSolvationEnergy
OEFF::OEComplexEnergies, 150
GetComponents
OEMolPotential::OEGenericFF, 48
OEMolPotential::OEGenericFF2, 50
GetContactDistance
OEMolPotential::OEFFParams, 38
GetCoulombParams
OEMolPotential::OEFFParams, 39
GetCutoff
OEMolPotential::OENonBondIntcsOptions, 80
GetDegeneracy
OEMolSmirnoff::OESmirnoffTorsionData, 132
GetDesolvationEnergy
OEFF::OEComplexEnergies, 150
GetDielectric
OEFF::OEMMFFSheffieldOptions, 167
GetDielectricDerivative
OESheff::OESheffield, 137
GetEneConst
OEMolSmirnoff::OESmirnoffBendData, 131
OEMolSmirnoff::OESmirnoffImproperData, 133
OEMolSmirnoff::OESmirnoffStretchData, 130
OEMolSmirnoff::OESmirnoffTorsionData, 132
GetEpsilon
OEMolSmirnoff::OESmirnoffVdWData, 135
GetEquilibriumAngle
OEMolPotential::OEFFParams, 39
OEMolSmirnoff::OESmirnoffBendData, 131
GetEquilibriumBondLength
OEMolPotential::OEFFParams, 39
OEMolSmirnoff::OESmirnoffStretchData, 130
GetExponent
OEFF::OEMMFFSheffieldOptions, 167
GetFComponents
OEOpt::OEFunc0, 20
GetForceFieldType
OEFF::OEMMFFSheffieldOptions, 167
GetHessianDimension
OEOpt::OEBFGSOpt, 13
GetHostComponents
OEFF::OEComplexEnergies, 150
GetHostEnergy
OEFF::OEComplexEnergies, 150
GetHostSolvationEnergy
OEFF::OEComplexEnergies, 150
GetID
OEMolSmirnoff::OESmirnoffData, 129
GetIEFFTerms
OEFF::OEMMFFIEFFOptions, 164
GetImproperData
OEMolSmirnoff::OESmirnoffParams, 127
GetInteractions
OEMolPotential::OEMolFunc, 68
GetInterComponents
OEFF::OEComplexEnergies, 150
GetInterEnergy
OEFF::OEComplexEnergies, 151
GetIntermolecularVdWCutoff
OEFF::OEMMFFAmberOptions, 161
GetInverseHessian
OEOpt::OEBFGSOpt, 13
GetIterLimit
OEOpt::OEOptimizer0, 28
OEOpt::OEOptimizer1, 29
OEOpt::OEOptimizer2, 30
GetKeepHessian
OEOpt::OEBFGSOpt, 13
GetLigandComponents
OEFF::OEComplexEnergies, 151
GetLigandEnergy
OEFF::OEComplexEnergies, 151
GetLigandFF
OEFF::OEComplexFF, 152
GetLigandSolvationEnergy
OEFF::OEComplexEnergies, 151
GetList
OEMolPotential::OEInteractParams, 53
GetMMFFTerms
OEFF::OEMMFFAmberOptions, 162
OEFF::OEMMFFIEFFOptions, 164
GetName
OEMolPotential::OEInteraction, 51
GetNonBondOptions
OEMolPotential::OENonBondPotBase, 78
GetOutOfPlaneParams
OEMolPotential::OEFFParams, 42
GetPenaltyConstant
OEConstr::OEDihedralQuadPenalty, 109
GetPeriodicity
OEMolSmirnoff::OESmirnoffImproperData, 134
OEMolSmirnoff::OESmirnoffTorsionData, 132
GetPhaseAngle
OEMolSmirnoff::OESmirnoffImproperData, 134
OEMolSmirnoff::OESmirnoffTorsionData, 132
GetRadius

OEMolSmirnoff::OESmirnoffVdWData, 135
GetSmarts
OEMolSmirnoff::OESmirnoffData, 129
GetSoluteDielectricConstant
OESheff::OESheffieldOptions, 138
GetSolventDielectricConstant
OESheff::OESheffieldOptions, 138
GetSpecifiedAngle
OEConstr::OEDihedralQuadPenalty, 109
GetStretchBendParams
OEMolPotential::OEFFParams, 39
GetStretchData
OEMolSmirnoff::OESmirnoffParams, 127
GetStretchParams
OEMolPotential::OEFFParams, 40
GetTitle
OEMolPotential::OEFFParams, 43
OEMolPotential::OEForceField, 46
GetTolerance
OEOpt::OEOptimizer1, 29
OEOpt::OEOptimizer2, 30
GetTorsionData
OEMolSmirnoff::OESmirnoffParams, 127
GetTorsionParams
OEMolPotential::OEFFParams, 41
GetTotalEnergy
OEFF::OEComplexEnergies, 151
GetUseCoulomb
OEMolPotential::OENonBondIntcsOptions, 80
GetUseExternalCharges
OEFF::OEMMFFAmberOptions, 162
GetUseMMFF94s
OEFF::OEMMFFAmberOptions, 162
OEFF::OEMMFFIEFFOptions, 164
GetUseVdW
OEMolPotential::OENonBondIntcsOptions, 80
GetValues
OEMolPotential::OEInteraction, 52
GetVar
OEFF::OEComplexFF, 153
OEMolPotential::OEInterMolFunc1, 57
OEOpt::OEAdaptor, 12
GetVdWData
OEMolSmirnoff::OESmirnoffParams, 127
GetVdwParams
OEMolPotential::OEFFParams, 40
GetVdwRadius
OEMolPotential::OEFFParams, 39

H

HasBendParams
OEMolPotential::OEFFParams, 42
HasCoulomb
OEMolPotential::OEForceField, 46
HasInternalCharges
OEMolPotential::OEForceField, 46
HasStretchParams
OEMolPotential::OEFFParams, 42
HasTorsionParams
OEMolPotential::OEFFParams, 43
HasVdwParams
OEMolPotential::OEFFParams, 42

I

IsAngle
OEMolPotential::OEBendParams, 34
IsBond
OEMolPotential::OEStretchParams, 89
IsPair
OEMolPotential::OEVdwParams, 100
IsRedundant
OEMolPotential::OEQuatAdaptor, 86
IsTorsion
OEMolPotential::OETorsionParams, 98
IsValid
OEMolPotential::OEBendParams, 34
OEMolPotential::OEOutOfPlaneParams, 83
OEMolPotential::OEStretchParams, 89
OEMolPotential::OETorsionParams, 97
OEMolPotential::OEVdwParams, 100

L

Load
OEFF::OEFF14SBSmirnoffParams, 149
OEMolSmirnoff::OESmirnoffParams, 128

N

NumAtoms
OEMolPotential::OEInteraction, 52
NumTerms
OEMolSmirnoff::OESmirnoffTorsionData, 132
NumValues
OEMolPotential::OEInteraction, 52
NumVar
OEOpt::OEFunc0, 20

O

OEConstr::OEDihedralQuadPenalty, 108
Constructors, 109
GetPenaltyConstant, 109
GetSpecifiedAngle, 109
OEConstr::OEHarmonicPotential, 109
Constructors, 110
Get_dc, 111
Get_kc, 111
Set, 111
Set_dc, 111

Set_kc, 111
OEFF::OEClearForceFieldDummyAtom, 169
OEFF::OEClearForceFieldDummyAtoms, 169
OEFF::OEComplexEnergies, 149
Constructors, 150
GetComplexSolvationEnergy, 150
GetDesolvationEnergy, 150
GetHostComponents, 150
GetHostEnergy, 150
GetHostSolvationEnergy, 150
GetInterComponents, 150
GetInterEnergy, 151
GetLigandComponents, 151
GetLigandEnergy, 151
GetLigandSolvationEnergy, 151
GetTotalEnergy, 151
OEFF::OEComplexFF, 151
AdaptHostCoords, 152
Constructors, 152
GetLigandFF, 152
GetVar, 153
Set, 153
SetHostFlex, 153
SetNonBondCutoff, 153
SetupHost, 153
OEFF::OEComplexFFParameter, 154
Constructors, 155
OEFF::OEFF14SBParsley, 139
Constructors, 140
OEFF::OEFF14SBParsleyComplex, 140
Constructors, 141
OEFF::OEFF14SBParsleyParams, 141
Constructors, 143
OEFF::OEFF14SBSage, 143
Constructors, 144
OEFF::OEFF14SBSageComplex, 144
Constructors, 144
OEFF::OEFF14SBSageParams, 145
Constructors, 146
OEFF::OEFF14SBSmirnoff, 146
Constructors, 147
PrepMol, 147
OEFF::OEFF14SBSmirnoffComplex, 147
Constructors, 148
OEFF::OEFF14SBSmirnoffParams, 148
Constructors, 149
Load, 149
OEFF::OEFFGetArch, 171
OEFF::OEFFGetLicensee, 171
OEFF::OEFFGetPlatform, 171
OEFF::OEFFGetRelease, 171
OEFF::OEFFGetSite, 172
OEFF::OEFFGetVersion, 172
OEFF::OEFFIsLicensed, 172
OEFF::OEForceFieldParameter, 155
Constructors, 156
OEFF::OEGetMMFFSheffieldFFName, 169
OEFF::OEGetMMFFSheffieldFFType, 170
OEFF::OEIsForceFieldDummyAtom, 170
OEFF::OEIsValidMMFFSheffieldFF, 170
OEFF::OELigandFFType, 172
OEFF::OELigandFFType::MMFF94, 172
OEFF::OELigandFFType::MMFF94S, 172
OEFF::OELigandFFType::PARSLEY, 173
OEFF::OELigandFFType::SAGE, 173
OEFF::OEMMFFAmber, 158
Constructors, 159
PrepMol, 159
Setup, 159
OEFF::OEMMFFAmberComplex, 159
Constructors, 160
OEFF::OEMMFFAmberOptions, 160
Constructors, 161
CreateCopy, 161
GetAmberTerms, 161
GetIntermolecularVdWCutoff, 161
GetMMFFTerms, 162
GetUseExternalCharges, 162
GetUseMMFF94s, 162
SetAmberTerms, 162
SetIntermolecularVdWCutoff, 162
SetMMFFTerms, 162
SetUseExternalCharges, 162
SetUseMMFF94s, 162
OEFF::OEMMFFComplex, 163
Constructors, 163
OEFF::OEMMFFIEFF, 156
Constructors, 158
OEFF::OEMMFFIEFFOptions, 163
Constructors, 164
GetIEFFTerms, 164
GetMMFFTerms, 164
GetUseMMFF94s, 164
SetIEFFTerms, 164
SetMMFFTerms, 164
SetUseMMFF94s, 165
OEFF::OEMMFFSheffield, 165
Constructors, 166
OEFF::OEMMFFSheffieldFFType, 173
OEFF::OEMMFFSheffieldFFType::Invalid, 174
OEFF::OEMMFFSheffieldFFType::MMFF, 173
OEFF::OEMMFFSheffieldFFType::MMFF94S, 173
OEFF::OEMMFFSheffieldFFType::MMFF94S_NOESTAT, 173
OEFF::OEMMFFSheffieldFFType::MMFF94S_SHEFF, 174
OEFF::OEMMFFSheffieldFFType::MMFF94S_TRUNC, 173

OEFF::OEMMFFSheffieldFFType::MMFF94Smod, 174
OEFF::OEMMFFSheffieldFFType::MMFF94Smod_NOESTAT, 174
OEFF::OEMMFFSheffieldFFType::MMFF94Smod_SHEFF, 174
OEFF::OEMMFFSheffieldFFType::MMFF94Smod_TRUNC, 174
OEFF::OEMMFFSheffieldFFType::MMFF_NOESTAT, 173
OEFF::OEMMFFSheffieldFFType::MMFF_SHEFF, 173
OEFF::OEMMFFSheffieldFFType::MMFF_TRUNC, 173
OEFF::OEMMFFSheffieldOptions, 166
    Constructors, 167
    GetDielectric, 167
    GetExponent, 167
    GetForceFieldType, 167
    SetDielectric, 167
    SetExponent, 167
    SetForceFieldType, 167
OEFF::OESetForceFieldDummyAtom, 170
OEFF::OESmirnoffComplex, 167
    Constructors, 168
OEMolMMFF::OEIsValidMMFFMolecule, 118
OEMolMMFF::OEMMFF, 111
    Constructors, 113
OEMolMMFF::OEMMFF94sParams, 113
    Constructors, 114
OEMolMMFF::OEMMFFCalcVdw, 118
OEMolMMFF::OEMMFFOutOfPlane, 114
    Constructors, 115
OEMolMMFF::OEMMFFParams, 116
    Constructors, 117
OEMolMMFF::OEMMFFStretchBend, 117
    Constructors, 118
    Set, 118
OEMolPotential::OEBendParams, 33
    Constructors, 34
    GetAtomIndex1, 34
    GetAtomIndex2, 34
    GetAtomIndex3, 34
    IsAngle, 34
    IsValid, 34
OEMolPotential::OEBendPotential, 35
    Assignment operator, 35
    Constructors, 35
    Set, 36
OEMolPotential::OECoulombPotential, 36
    Assignment operator, 37
    Constructors, 36
OEMolPotential::OEFFParam, 101
OEMolPotential::OEFFParam::a0, 103
OEMolPotential::OEFFParam::BendConst, 103
OEMolPotential::OEFFParam::ContactDist, 102
OEMolPotential::OEFFParam::eps, 103
OEMolPotential::OEFFParam::Epsilon, 102
OEMolPotential::OEFFParam::EqAngle, 103
OEMolPotential::OEFFParam::EqDist, 102
OEMolPotential::OEFFParam::EqDistance, 102
OEMolPotential::OEFFParam::k, 102
OEMolPotential::OEFFParam::ka, 103
OEMolPotential::OEFFParam::kb1, 103
OEMolPotential::OEFFParam::kb2, 103
OEMolPotential::OEFFParam::ko, 104
OEMolPotential::OEFFParam::listDv, 105
OEMolPotential::OEFFParam::listKv, 105
OEMolPotential::OEFFParam::listPhv, 105
OEMolPotential::OEFFParam::listPv, 105
OEMolPotential::OEFFParam::ntt, 105
OEMolPotential::OEFFParam::OOP, 104
OEMolPotential::OEFFParam::OutOfPlaneConst, 104
OEMolPotential::OEFFParam::p1, 105
OEMolPotential::OEFFParam::ph1, 105
OEMolPotential::OEFFParam::r0, 102
OEMolPotential::OEFFParam::r1, 103
OEMolPotential::OEFFParam::r2, 104
OEMolPotential::OEFFParam::rij, 102
OEMolPotential::OEFFParam::StretchBendConst1, 103
OEMolPotential::OEFFParam::StretchBendConst2, 103
OEMolPotential::OEFFParam::StretchConst, 102
OEMolPotential::OEFFParam::V1, 104
OEMolPotential::OEFFParam::v1, 104
OEMolPotential::OEFFParam::V2, 104
OEMolPotential::OEFFParam::v2, 104
OEMolPotential::OEFFParam::V3, 104
OEMolPotential::OEFFParam::v3, 104
OEMolPotential::OEFFParams, 37
    Assignment operator, 38
    Constructors, 38
    CreateCopy, 38
    GetAtomTypeIndex, 38
    GetAtomTypeName, 38
    GetBendParams, 41
    GetContactDistance, 38
    GetCoulombParams, 39
    GetEquilibriumAngle, 39
    GetEquilibriumBondLength, 39
    GetOutOfPlaneParams, 42
    GetStretchBendParams, 39
    GetStretchParams, 40
    GetTitle, 43
    GetTorsionParams, 41
    GetVdwParams, 40
    GetVdwRadius, 39
    HasBendParams, 42
    HasStretchParams, 42

HasTorsionParams, 43
HasVdwParams, 42
PrepMol, 39
OEMolPotential::OEFFPotential, 43
Assignment operator, 44
Constructors, 44
Set, 44
OEMolPotential::OEForceField, 44
GetTitle, 46
HasCoulomb, 46
HasInternalCharges, 46
PrepMol, 46
Set, 46
SetNonBondCutoff, 47
SetTitle, 47
OEMolPotential::OEFuncType, 106
OEMolPotential::OEFuncType::All, 107
OEMolPotential::OEFuncType::AllRepel, 107
OEMolPotential::OEFuncType::Bend, 107
OEMolPotential::OEFuncType::Bond, 106
OEMolPotential::OEFuncType::Coulomb, 106
OEMolPotential::OEFuncType::HarmonicPot, 108
OEMolPotential::OEFuncType::ImproperTorsion, 107
OEMolPotential::OEFuncType::InterAdapt, 108
OEMolPotential::OEFuncType::InterMolCoulomb, 107
OEMolPotential::OEFuncType::InterMolVdW, 107
OEMolPotential::OEFuncType::OutOfPlane, 107
OEMolPotential::OEFuncType::QuatAdapt, 108
OEMolPotential::OEFuncType::Sheffield, 108
OEMolPotential::OEFuncType::Stretch, 107
OEMolPotential::OEFuncType::StretchBend, 107
OEMolPotential::OEFuncType::SubsetAdapt, 108
OEMolPotential::OEFuncType::SubsetAdaptor, 108
OEMolPotential::OEFuncType::TorAdapt, 108
OEMolPotential::OEFuncType::Torsion, 107
OEMolPotential::OEFuncType::Undefined, 106
OEMolPotential::OEFuncType::VdW, 106
OEMolPotential::OEFuncType::VdWRepel, 106
OEMolPotential::OEGenericFF, 47
AddMolFunc, 48
Constructors, 48
GetComponents, 48
RemoveMolFunc, 49
OEMolPotential::OEGenericFF2, 49
AddMolFunc, 50
Constructors, 50
GetComponents, 50
RemoveMolFunc, 51
OEMolPotential::OEInteraction, 51
Constructors, 51
GetAtoms, 51
GetName, 51
GetValues, 52
NumAtoms, 52
NumValues, 52
OEMolPotential::OEInteractParams, 52
Assignment operator, 53
Clear, 53
Constructors, 52
Get, 53
GetList, 53
Set, 53
SetList, 53
OEMolPotential::OEInterAdaptor, 53
AdaptGrad, 55
Constructors, 55
Set, 55
OEMolPotential::OEInterCoulombPotential, 55
Assignment operator, 56
Constructors, 56
OEMolPotential::OEInterMolFunc1, 56
AdaptHostCoords, 57
GetVar, 57
SetHostFlex, 57
SetupHost, 57
OEMolPotential::OEInterMolFunc2, 58
OEMolPotential::OEInterNonBondPotBase, 58
Assignment operator, 60
Constructors, 59
Set, 60
SetNonBondCutoff, 60
OEMolPotential::OEInterNonBondPotential, 60
Assignment operator, 61
Constructors, 61
OEMolPotential::OEInterVdwPotential, 61
Assignment operator, 62
Constructors, 62
OEMolPotential::OEMolAdaptor, 62
SetMolFunc, 64
OEMolPotential::OEMolAdaptor2, 64
SetMolFunc2, 65
OEMolPotential::OEMolFunc, 65
Clear, 68
ClearPredicates, 68
GetInteractions, 68
Set, 68
Setup, 69
SetupConf, 69
OEMolPotential::OEMolFunc0, 69
OEMolPotential::OEMolFunc1, 72
OEMolPotential::OEMolFunc2, 75
OEMolPotential::OENonBondIntcsOptions, 79
Assignment operator, 79
Constructors, 79
CreateCopy, 80

GetCutoff, 80
GetUseCoulomb, 80
GetUseVdW, 80
SetCutoff, 80
SetUseCoulomb, 80
SetUseVdW, 80
OEMolPotential::OENonBondPotBase, 76
Assignment operator, 77
Constructors, 77
GetNonBondOptions, 78
Set, 78
OEMolPotential::OENonBondPotential, 78
Assignment operator, 79
Constructors, 79
OEMolPotential::OENumericMolFunc2, 80
Constructors, 81
Set, 82
OEMolPotential::OEOutOfPlaneParams, 82
Constructors, 82
GetAtomIndex1, 83
GetAtomIndex2, 83
GetAtomIndex3, 83
GetAtomIndex4, 83
IsValid, 83
OEMolPotential::OEOutOfPlanePotential, 83
Assignment operator, 84
Constructors, 84
OEMolPotential::OEQuatAdaptor, 84
AdaptGrad, 86
Constructors, 85
IsRedundant, 86
OEMolPotential::OEScaledMolFunc, 86
Constructors, 87
Set, 88
OEMolPotential::OEStretchParams, 88
Constructors, 88
GetAtomIndex1, 89
GetAtomIndex2, 89
IsBond, 89
IsValid, 89
OEMolPotential::OEStretchPotential, 89
Assignment operator, 90
Constructors, 90
Set, 90
OEMolPotential::OESubsetAdaptor, 90
AdaptGrad, 92
AdaptHessian, 92
Constructors, 92
Set, 92
OEMolPotential::OETorAdaptor, 92
AdaptGrad, 94
Constructors, 94
Set, 94
OEMolPotential::OETorQuatAdaptor, 94
AdaptGrad, 96
Constructors, 95
Set, 96
OEMolPotential::OETorsionParams, 96
Constructors, 97
GetAtomIndex1, 97
GetAtomIndex2, 97
GetAtomIndex3, 97
GetAtomIndex4, 97
IsTorsion, 98
IsValid, 97
OEMolPotential::OETorsionPotential, 98
Assignment operator, 98
Constructors, 98
Set, 99
OEMolPotential::OEVarType, 105
OEMolPotential::OEVarType::Cart, 106
OEMolPotential::OEVarType::SolidBody, 105
OEMolPotential::OEVarType::Torsions, 106
OEMolPotential::OEVarType::TranRotTor, 106
OEMolPotential::OEVdwParams, 99
Constructors, 99
GetAtom1, 100
GetAtom2, 100
GetAtomIndex1, 99
GetAtomIndex2, 100
IsPair, 100
IsValid, 100
OEMolPotential::OEVdwPotential, 100
Assignment operator, 101
Constructors, 101
OEMolSmirnoff::OEParsley, 119
Constructors, 120
OEMolSmirnoff::OEParsleyParams, 120
Constructors, 121
OEMolSmirnoff::OESage, 121
Constructors, 122
OEMolSmirnoff::OESageParams, 122
Constructors, 123
OEMolSmirnoff::OESmirnoff, 123
Constructors, 124
OEMolSmirnoff::OESmirnoffBendData, 130
Constructors, 131
GetEneConst, 131
GetEquilibriumAngle, 131
SetParams, 131
OEMolSmirnoff::OESmirnoffData, 128
GetID, 129
GetSmarts, 129
OEMolSmirnoff::OESmirnoffImproperData, 133
Constructors, 133
GetEneConst, 133
GetPeriodicity, 134
GetPhaseAngle, 134

SetParams, 134
OEMolSmirnoff::OESmirnoffParams, 124
AddBendData, 126
AddImproperData, 126
AddStretchData, 126
AddTorsionData, 126
AddVdWData, 126
Constructors, 126
GetBendData, 127
GetImproperData, 127
GetStretchData, 127
GetTorsionData, 127
GetVdWData, 127
Load, 128
SetBendData, 128
SetImproperData, 128
SetStretchData, 128
SetTorsionData, 128
SetVdWData, 128
OEMolSmirnoff::OESmirnoffStretchData, 129
Constructors, 130
GetEneConst, 130
GetEquilibriumBondLength, 130
SetParams, 130
OEMolSmirnoff::OESmirnoffTorsionData, 131
AddData, 132
ClearData, 133
Constructors, 132
GetDegeneracy, 132
GetEneConst, 132
GetPeriodicity, 132
GetPhaseAngle, 132
NumTerms, 132
OEMolSmirnoff::OESmirnoffType, 135
OEMolSmirnoff::OESmirnoffType::PARSLEY_OPENFF, 135
OEMolSmirnoff::OESmirnoffType::SAGE_OPENFF, 135
OEMolSmirnoff::OESmirnoffType::SMIRNOFF99FROSST, 135
OEMolSmirnoff::OESmirnoffVdWData, 134
Constructors, 134
GetEpsilon, 135
GetRadius, 135
SetParams, 135
OEOpt::OEAdaptor, 11
AdaptVar, 12
GetVar, 12
OEOpt::OEBFGSOpt, 12
Constructors, 13
GetHessianDimension, 13
GetInverseHessian, 13
GetKeepHessian, 13
SetKeepHessian, 13
UseSavedHessian, 13
OEOpt::OECGLineMinimize, 13
Constructors, 14
OEOpt::OECheckpoint0, 14
operator(), 14
OEOpt::OECheckpoint1, 14
operator(), 15
OEOpt::OEConjGradOpt, 15
Constructors, 16
OEOpt::OEDFPOpt, 16
Constructors, 16
OEOpt::OEFComponent, 16
Constructors, 17
OEOpt::OEFletcherReevesOpt, 17
Constructors, 17
OEOpt::OEFunc0, 17
GetFComponents, 20
NumVar, 20
operator(), 20
OEOpt::OEFunc1, 20
operator(), 23
OEOpt::OEFunc2, 23
operator(), 25
OEOpt::OEHestenesStiefelOpt, 25
Constructors, 25
OEOpt::OELineMinimize, 26
CreateCopy, 26
operator(), 26
SetMaxStep, 26
OEOpt::OEMaxStepLineMinimize, 27
Constructors, 27
OEOpt::OENewtonOpt, 33
Constructors, 33
OEOpt::OENRLineSearch, 27
Constructors, 27
OEOpt::OEOptimizer0, 27
GetIterLimit, 28
operator(), 28
SetIterLimit, 28
OEOpt::OEOptimizer1, 28
GetIterLimit, 29
GetTolerance, 29
operator(), 29
SetIterLimit, 29
SetLineMinimize, 29
SetTolerance, 30
OEOpt::OEOptimizer2, 30
GetIterLimit, 30
GetTolerance, 30
operator(), 30
SetIterLimit, 31
SetTolerance, 31
OEOpt::OEQNLineMinimize, 31
Constructors, 31

OEOpt::OERibierePolakOpt, 31
    Constructors, 32
OEOpt::OESteepestDescentOpt, 32
    Constructors, 32
    SetInitialStep, 32
OESheff::OESheffield, 135
    Constructors, 137
    GetDielectricDerivative, 137
    Set, 137
    SetCharges, 137
    SetRadii, 137
OESheff::OESheffieldOptions, 137
    Constructors, 138
    GetA, 138
    GetB, 138
    GetC, 138
    GetSoluteDielectricConstant, 138
    GetSolventDielectricConstant, 138
    SetA, 138
    SetB, 139
    SetC, 139
    SetSoluteDielectricConstant, 139
    SetSolventDielectricConstant, 139
OESheff::OESheffieldParamType, 139
OESheff::OESheffieldParamType::Protein, 139
OESheff::OESheffieldParamType::SmallMol, 139
operator()
    OEOpt::OECheckpoint0, 14
    OEOpt::OECheckpoint1, 15
    OEOpt::OEFunc0, 20
    OEOpt::OEFunc1, 23
    OEOpt::OEFunc2, 25
    OEOpt::OELineMinimize, 26
    OEOpt::OEOptimizer0, 28
    OEOpt::OEOptimizer1, 29
    OEOpt::OEOptimizer2, 30

P

PrepMol
    OEFF::OEFF14SBSmirnoff, 147
    OEFF::OEMMFFAmber, 159
    OEMolPotential::OEFFParams, 39
    OEMolPotential::OEForceField, 46

R

RemoveMolFunc
    OEMolPotential::OEGenericFF, 49
    OEMolPotential::OEGenericFF2, 51

S

Set
    OEConstr::OEHarmonicPotential, 111
    OEFF::OEComplexFF, 153
    OEMolMMFF::OEMMFFStretchBend, 118
    OEMolPotential::OEBendPotential, 36
    OEMolPotential::OEFFPotential, 44
    OEMolPotential::OEForceField, 46
    OEMolPotential::OEInteractParams, 53
    OEMolPotential::OEInterAdaptor, 55
    OEMolPotential::OEInterNonBondPotBase, 60
    OEMolPotential::OEMolFunc, 68
    OEMolPotential::OENonBondPotBase, 78
    OEMolPotential::OENumericMolFunc2, 82
    OEMolPotential::OEScaledMolFunc, 88
    OEMolPotential::OEStretchPotential, 90
    OEMolPotential::OESubsetAdaptor, 92
    OEMolPotential::OETorAdaptor, 94
    OEMolPotential::OETorQuatAdaptor, 96
    OEMolPotential::OETorsionPotential, 99
    OESheff::OESheffield, 137
Set_dc
    OEConstr::OEHarmonicPotential, 111
Set_kc
    OEConstr::OEHarmonicPotential, 111
SetA
    OESheff::OESheffieldOptions, 138
SetAmberTerms
    OEFF::OEMMFFAmberOptions, 162
SetB
    OESheff::OESheffieldOptions, 139
SetBendData
    OEMolSmirnoff::OESmirnoffParams, 128
SetC
    OESheff::OESheffieldOptions, 139
SetCharges
    OESheff::OESheffield, 137
SetCutoff
    OEMolPotential::OENonBondIntcsOptions, 80
SetDielectric
    OEFF::OEMMFFSheffieldOptions, 167
SetExponent
    OEFF::OEMMFFSheffieldOptions, 167
SetForceFieldType
    OEFF::OEMMFFSheffieldOptions, 167
SetHostFlex
    OEFF::OEComplexFF, 153
    OEMolPotential::OEInterMolFunc1, 57
SetIEFFTerms
    OEFF::OEMMFFIEFFOptions, 164
SetImproperData
    OEMolSmirnoff::OESmirnoffParams, 128
SetInitialStep
    OEOpt::OESteepestDescentOpt, 32
SetIntermolecularVdWCutoff
    OEFF::OEMMFFAmberOptions, 162
SetIterLimit
    OEOpt::OEOptimizer0, 28
    OEOpt::OEOptimizer1, 29

OEOpt::OEOptimizer2, 31
SetKeepHessian
OEOpt::OEBFGSOpt, 13
SetLineMinimize
OEOpt::OEOptimizer1, 29
SetList
OEMolPotential::OEInteractParams, 53
SetMaxStep
OEOpt::OELineMinimize, 26
SetMMFFTerms
OEFF::OEMMFFAmberOptions, 162
OEFF::OEMMFFIEFFOptions, 164
SetMolFunc
OEMolPotential::OEMolAdaptor, 64
SetMolFunc2
OEMolPotential::OEMolAdaptor2, 65
SetNonBondCutoff
OEFF::OEComplexFF, 153
OEMolPotential::OEForceField, 47
OEMolPotential::OEInterNonBondPotBase, 60
SetParams
OEMolSmirnoff::OESmirnoffBendData, 131
OEMolSmirnoff::OESmirnoffImproperData, 134
OEMolSmirnoff::OESmirnoffStretchData, 130
OEMolSmirnoff::OESmirnoffVdWData, 135
SetRadii
OESheff::OESheffield, 137
SetSoluteDielectricConstant
OESheff::OESheffieldOptions, 139
SetSolventDielectricConstant
OESheff::OESheffieldOptions, 139
SetStretchData
OEMolSmirnoff::OESmirnoffParams, 128
SetTitle
OEMolPotential::OEForceField, 47
SetTolerance
OEOpt::OEOptimizer1, 30
OEOpt::OEOptimizer2, 31
SetTorsionData
OEMolSmirnoff::OESmirnoffParams, 128
Setup
OEFF::OEMMFFAmber, 159
OEMolPotential::OEMolFunc, 69
SetupConf
OEMolPotential::OEMolFunc, 69
SetupHost
OEFF::OEComplexFF, 153
OEMolPotential::OEInterMolFunc1, 57
SetUseCoulomb
OEMolPotential::OENonBondIntcsOptions, 80
SetUseExternalCharges
OEFF::OEMMFFAmberOptions, 162
SetUseMMFF94s
OEFF::OEMMFFAmberOptions, 162
OEFF::OEMMFFIEFFOptions, 165
SetUseVdW
OEMolPotential::OENonBondIntcsOptions, 80
SetVdWData
OEMolSmirnoff::OESmirnoffParams, 128

U

UseSavedHessian
OEOpt::OEBFGSOpt, 13